

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## ZÁZNAMOVÁ APLIKACE MNEMOSYNE

LOGGING APPLICATION MNEMOSYNE

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Jaroslav Pizur

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Stanislav Pikula

BRNO 2019

# Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Jaroslav Pizur

**ID:** 186164

**Ročník:** 3

**Akademický rok:** 2018/19

**NÁZEV TÉMATU:**

## Záznamová aplikace Mnemosyne

### POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvoření logovacího programu v prostředí LabVIEW pro využití s univerzální měřicí kartou NI 9219. Doporučená osnova práce:

- 1) Stručně uveďte programovací prostředí LabVIEW, popište univerzální kartu NI 9219 a její možnosti.
- 2) Navrhněte univerzální logovací aplikaci cílenou na DAQ kartu NI 9219. Program navrhněte tak, aby umožňoval měřit veličiny všemi způsoby, které NI 9219 umožňuje. Zvažte asynchronní architekturu pro vyčítání dat a logování. Uvažujte výběr parametrů měření uživatelsky přívětivým způsobem a tak, aby bylo možné na každém ze čtyř vstupů karty měřit jinou veličinu s jiným nastavením. Data ukládejte do TDMS formátu a zajistěte, že do hlavičky TDMS souboru se ukládají všechny parametry nastavení, aby bylo měření v rámci uloženého souboru plně zdokumentováno. Umožněte načíst nastavení vstupů na základě dříve uloženého TDMS souboru tak, aby bylo možné zopakovat měření pouhým načtením dříve uložených dat.
- 3) Realizujte navrženou aplikaci v programovacím prostředí LabVIEW. Kód programu vhodně dokumentujte. V textu práce zdokumentujte uživatelské rozhraní a funkčnost.
- 4) Připravte program na využití jako samostatnou spustitelnou aplikaci.
- 5) Otestujte plnou funkčnost programu na ukázkovém měření více veličin zároveň.

### DOPORUČENÁ LITERATURA:

VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. Zacináme s LabVIEW. Praha: BEN - technická literatura, 2008. ISBN 978-80-7300-245-9.

**Termín zadání:** 4.2.2019

**Termín odevzdání:** 20.5.2019

**Vedoucí práce:** Ing. Stanislav Pikula

**Konzultant:**

**doc. Ing. Václav Jirsík, CSc.**  
předseda oborové rady

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Cieľom tejto bakalárskej práce je vytvorenie aplikácie s názvom Mnemosyne, ktorá bude schopná merať na každom zo štyroch kanálov karty NI9219 iný typ merania. Aplikácia vie namerané dáta logovať do TDMS súboru a zároveň do hlavičky daného súboru uloží všetky potrebné nastavenia a parametre pre umožnenie kompletného zreprodukovania akéhokoľvek predchádzajúceho merania vyčítaním týchto nastavení a parametrov zo zvoleného súboru. V závere práce sa tieto požadované vlastnosti aplikácie otestujú s fyzickou kartou NI9219 na meraní viacerých veličín zároveň.

## KĽÚČOVÉ SLOVÁ

NI 9219, logovacia aplikácia, C-Series, USB-9162, cDAQ-9171, LabVIEW, G programovací jazyk, zber dát, TDMS, NI-DAQmx, DAQ, CompactDAQ

## ABSTRACT

The goal of this bachelor's thesis is creation of an application called Mnemosyne, which will be able to measure different types of measurement on each of NI 9219 four input channels. The application can log the data to a TDMS file and at the same time is able to write parameters and settings to its header. Thus every measurement is fully documented and application is able to reproduce any measurement by loading the header of any previously created file. The application is then tested for two different types of measurement on two distinct channels.

## KEYWORDS

NI 9219, logging application, C-Series, USB-9162, cDAQ-9171, LabVIEW, G programming language, data acquisition, TDMS, NI-DAQmx, DAQ, CompactDAQ

PIZUR, Jaroslav. *Záznamová aplikace Mnemosyne*. Brno, 2019, 79 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedúci práce: Ing. Stanislav Pikula

## VYHLÁSENIE

Vyhlasujem, že som svoju bakalársku prácu na tému „Záznamová aplikace Mnemosyne“ vypracoval samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno .....

.....

podpis autora

## POĎAKOVANIE

Ďakujem vedúcemu bakalárskej práce pánovi Ing. Stanislavovi Pikulovi, predovšetkým za neustálu ochotu komunikovať a v akejkoľvek situácii v práci poradiť a poskytnúť vedenie.

Brno .....

.....

podpis autora

# Obsah

Úvod	10
<b>1 Teoretická časť práce</b>	<b>11</b>
1.1 Využitie meriacich kariet v HW od NI	11
1.1.1 PXI	11
1.1.2 CompactRIO	12
1.1.3 PC-Based Systems	12
1.2 Softwarové prostredia NI	13
1.2.1 Measurement Studio	13
1.2.2 LabWindows/CVI	13
1.3 LabVIEW	14
1.3.1 Výhody využitia LabVIEW	14
1.3.2 Popis LabVIEW	14
1.4 Možnosti meracej karty NI 9219	16
1.4.1 Napätie (Voltage)	20
1.4.2 Prúd (Current)	21
1.4.3 4-Vodičové meranie odporu (4-wire resistance) a Odporový teplomer RTD v 4-vodičovom zapojení	21
1.4.4 2-Vodičové meranie odporu (2-Wire resistance) a Štvrtinový mostík (Quarter-bridge)	22
1.4.5 Termoelektrický článok (Thermocouple)	22
1.4.6 Odporový teplomer (RTD) v 3-vodičovom zapojení	24
1.4.7 Polovičný mostík (Half-bridge)	25
1.4.8 Plný mostík (Full-bridge)	25
1.4.9 Digital In	26
1.4.10 Open Contact	26
<b>2 Návrh zhotovenia praktickej časti</b>	<b>27</b>
2.1 Návrh a popis UI	28
2.1.1 Pokročilý používateľ	28
2.1.2 Sprievodca	28
2.1.3 Hlavný užívateľský panel	29
2.1.4 Zmeny okien	29
2.2 Návrh a popis funkčnej časti programu	30
2.2.1 Riešenie clusteru so všetkými nastaveniami	30
2.2.2 Sekvenčná časť vo forme stavového automatu	31
2.2.3 Akvizícia dát	32

<b>3</b>	<b>Rozbor vhodnej šablóny</b>	<b>34</b>
3.1	Mechanizmus notifikátorov (Notifiers) . . . . .	34
3.2	Mechanizmus rady (Queue) . . . . .	36
3.3	User Event - udalosť . . . . .	37
3.4	Message queues . . . . .	39
<b>4</b>	<b>Výsledky študentskej práce</b>	<b>44</b>
4.1	Rozbor riešenia . . . . .	44
4.2	Mnemosyne . . . . .	44
4.2.1	Rozbor kódu . . . . .	44
4.2.2	Návod na použitie aplikácie . . . . .	51
4.3	Mnemasyn . . . . .	54
4.3.1	Rozbor kódu . . . . .	54
4.3.2	Návod na použitie aplikácie . . . . .	60
<b>5</b>	<b>Testovacie meranie</b>	<b>61</b>
5.1	Návrh . . . . .	61
5.2	Riešenie . . . . .	61
5.3	Výstup z merania . . . . .	66
<b>6</b>	<b>Možné rozšírenia programu</b>	<b>68</b>
6.1	Vzhľad . . . . .	68
6.2	Sprievodca nastaveniami - User Guide . . . . .	68
6.3	Zmeny pre hromadnú implementáciu . . . . .	68
<b>7</b>	<b>Záver</b>	<b>70</b>
	<b>Literatúra</b>	<b>72</b>
	<b>Zoznam symbolov, veličín a skratiek</b>	<b>78</b>



# Zoznam obrázkov

1.1	Základné obvodové schéma:[1]	18
1.2	TEDS zapojenie [2]	19
1.3	Schéma zobrazujúca zapojenie karty NI 9219 pre meranie napätia [1]	21
1.4	Schéma zapojenia kanálu karty NI 9219 pre meranie prúdu [6]	21
1.5	Schéma zapojenia kanálu karty NI 9219 pre 4-vodičové meranie odporu a odporový teplomer v 4-vodičovom zapojení [1]	22
1.6	Schéma zapojenia kanálu karty NI 9219 pre 2-vodičové meranie odporu a štvrtinový mostík [1]	23
1.7	Kompenzácia studeného konca [26]	24
1.8	Schéma zapojenia kanálu karty NI 9219 pre termoelektrický článok [1]	24
1.9	Schéma zapojenia kanálu karty NI 9219 pre odporový teplomer v 3-vodičovom zapojení [1]	25
1.10	Schéma zapojenia kanálu karty NI 9219 pre polovičný mostík [1]	25
1.11	Schéma zapojenia kanálu karty NI 9219 pre plný mostík [1]	26
1.12	Schéma zapojenia kanálu karty NI 9219 pre Digital In [1]	26
1.13	Schéma zapojenia kanálu karty NI 9219 pre Open Contact [1]	26
2.1	Blokové schéma pre plánované UI	27
2.2	Funkčné rozdelenie bodu 9 z Obr.č.2.1	32
3.1	funkcia „Obtain Notifier“	34
3.2	funkcia „funkcia Send Notification“	35
3.3	funkcia Wait on Notification	35
3.4	funkcia Release Notifier	35
3.5	Obtain Queue	36
3.6	Enqueue Element	36
3.7	Dequeue Element	37
3.8	Flush Queue	37
3.9	Release Queue	37
3.10	Vytvorenie udalosti (Event)	38
3.11	Mechanizmus „Dynamic registration“	38
3.12	Unregister For Events	38
3.13	Odoslanie udalosti	39
3.14	Odstránenie udalosti	39
3.15	Vytvorenie riadiacej rady (Queue) pre prenos správ	40
3.16	Pridanie prvku do rady	40
3.17	Create All Message Queues.vi	40
3.18	Stavy jednotlivých štruktúr v akvizičnej a logovacej slučke	41
3.19	Stavy Event štruktúry a Message Cases užívateľského rozhrania	42

3.20	funkcia „Unbundle By Name“ rozvetvujúca referencie rád (Queues) .	42
3.21	funkcia na spojenie chybových reťazcov „Merge Errors“ . . . . .	42
4.1	Blokový diagram aplikácie Mnemosyne . . . . .	45
4.2	Voľba Measurement Group . . . . .	52
4.3	Ukážka Measurement Groups v TDMS súbore . . . . .	53
4.4	Ukážka riadiaceho prvku pre zobrazenie nastavení počas akvizície . .	53
4.5	Blokové schéma aplikácie Mnemasyn [47] . . . . .	54
4.6	Inicializácia a nastavenie referencií . . . . .	55
4.7	Nastavenie viditeľnosti parametrov MCS . . . . .	56
4.8	Načítanie parametrov z TDMS súboru . . . . .	56
4.9	Nastavenie odporúčaných hodnôt parametrov . . . . .	56
4.10	Vytvorenie virtuálnych kanálov . . . . .	57
4.11	Zápis Timing Settings, start task . . . . .	57
4.12	Dátová akvizícia . . . . .	58
4.13	Zastavenie akvizície . . . . .	58
4.14	Zápis parametrov do TDMS . . . . .	59
4.15	Zápis timing nastavení do TDMS . . . . .	59
4.16	Logovanie dát . . . . .	59
4.17	Ukončenie logovania . . . . .	59
5.1	Náhľad na zostavu testovacieho merania . . . . .	62
5.2	Front panel počas merania odporu (Resistance) (Mnemasyn) . . . . .	63
5.3	Meranie 2-vodičového odporu (2-wire resistance) pri 10°C až 60°C (Mnemasyn) . . . . .	64
5.4	Meranie pomocou termočlánku (Thermocouple); (ref. 10°C až 60°C) - Mnemasyn . . . . .	64
5.5	Front panel počas merania termočlánkom (Thermocouple) (Mnemo- syne) . . . . .	65
5.6	Meranie 2-vodičového odporu (2-wire resistance) pri 10°C až 60°C (Mnemosyne) . . . . .	67
5.7	Meranie pomocou termočlánku (Thermocouple); (ref. 10°C až 60°C) - Mnemosyne . . . . .	67

# Úvod

Návrh zadania tejto práce pramenil z potreby jednoduchšej cesty, ako využívať meraciu kartu NI9219 bez toho, aby užívateľ musel pri každom meraní vytvárať špecifický kus kódu pre konkrétne využitie. Druhou možnosťou bolo hľadanie už existujúceho kódu pre meranie, ktoré chce užívateľ vykonať a je celkom pravdepodobné, že by sa k takému kódu aj dostal, ale všetko si pýta čas a nie vždy užívateľ meria jeden typ merania, poprípade nie vždy má čas hľadať špecifické riešenia naprieč dostupným zdrojom. Ideálnym riešením je aplikácia, ktorá by zahrňovala všetky možnosti, ktoré dokáže meracia karta s ktorou sa pracuje. Čo do výsledku potom pozorujeme, že v rámci použitia karty, ktorú má užívateľ v ruke, nebude z jeho strany potrebná prakticky ani znalosť samotného jazyka G, poprípade prostredia LabVIEW. Napriek tomu bude schopný využívať všetky jej dostupné možnosti.

Riešeniu tejto úlohy sa bude venovať táto práca. Cieľom je vytvoriť aplikáciu, ktorá bude schopná užívateľsky prívetivým spôsobom sprístupniť výber typu merania na každom zo štyroch kanálov meracej karty, ako aj nastavenie parametrov každého z týchto meraní. Meranie na každom kanále môže byť iné, ako aj žiadne, poprípade sa môžu typy meraní opakovať a predovšetkým musia byť sprístupnené všetky možnosti karty NI9219. Aplikácia bude logovať dáta do zvoleného TDMS súboru a všetky nastavenia, ktorým bude podliehať to dané meranie s kartou, budú uložené do hlavičky TDMS súboru, kde boli logované dáta. Aplikácia bude schopná parametre zo súboru aj načítať a zhrnutím týchto faktov vyplýva, že užívateľ bude veľmi jednoducho schopný reprodukovať akékoľvek predchádzajúce meranie.

V teoretickom úvode práce sa budem venovať meracím možnostiam karty NI9219, programovaciemu prostrediu LabVIEW a pozornosť venujeme aj popísaniu samotnej karty vo všeobecnosti. Zhrnieme si možnosti, ktorými sa budeme môcť uberať v rámci štruktúry kódu, navrhujeme predbežné blokové schéma programu a zvážime implementáciu existujúcich kúskov kódu z oblasti dátovej akvizície, logovania, či merania. Kód v priebehu práce budem dokumentovať a mechanizmy používané v praktickej časti popisovať a vysvetľovať v jednotlivých kapitolách.

Po vytvorení aplikácie, kódu funkčného na simulovanej karte, ho otestujem s fyzickou kartou meraním viacerých veličín zároveň. Meraním overím funkčnosť programu a schopnosť aplikácie komunikovať s reálnym HW (Hardware). Meranie zdokumentujem a vyvediem záver plynúci ako z výsledkov merania, tak z celkového konceptu stavby aplikácie, ktorý sme zvolili.

# 1 Teoretická časť práce

## Spoločnosť National Instruments (NI)

Korporát navrhuje, vyrába a predáva hardware a software v oblasti merania a riadenia. Z dôvodov rýchlo vyvíjajúcej sa technológie a potreby obnovovania jednotlivých funkčných prvkov v priemysle vyrába modulárny hardware, kde aj tkvie filozofia tejto spoločnosti. Vývojári samostatného programovacieho prostredia menom LabVIEW, navrhnutého na prácu s ich HW, nazývajú platformy ktoré ponúkajú ako software-centric, teda zvyčajne ponúkajúcu veľmi výhodnú kompatibilitu práve pri práci s ich softwarom. Čo do konkurencie je kľúčovým súperom firma Keysight, pôvodne pod Agilent, ktorá vyvíja produkty vytvárajúce priamu konkurenciu radám od firmy National Instruments. Pre predstavu veľkosti stáleho zloženia firmy, ku koncu roku 2016 podľa výročnej správy [30] zamestnávala táto spoločnosť 7552 pracovníkov celosvetovo, čo sa týka vývoja utratiac v priemere 230 miliónov USD ročne na výzkumné účely. V tú dobu mala zaregistrovaných 858 U.S. patentov, 45 v zahraničných krajinách. Čo je kľúčové na týchto informáciách je práve to, že 264 zo spomenutých 858 U.S. patentov sa týka vývojového prostredia LabVIEW a jeho základných funkcií či jeho grafického programovacieho prístupu. Z hľadiska prospešnosti využívania tohto vývojového prostredia v meracom a riadiacom priemysle firma NI získava istú formu menšej výhody, či kontroly nad trhom.

## 1.1 Využitie meriacich kariet v HW od NI

Pristúpiť k HW produktom od NI sa dá z dôsledku širokej škály zariadení rôznymi spôsobmi od aplikácie, na ktorú produkt potrebujeme, cez komunikačné rozhranie, až po peňažnú sumu, ktorú sme ochotní investovať. Spoločnosť svoj HW v základe delí na tri základné skupiny: PXI, CompactRio a PC-Based Systems. To nám pomôže sa v jednoduchosti zorientovať a prínajmenšom určiť, že naša karta sa môže použiť v CompactRio chassis a ďalšie možnosti má v oblasti PC-Based Systems, čo detailnejšie rozoberieme v ďalších kapitolách.

### 1.1.1 PXI

Táto modulárna platforma kombinuje mainstreamový PC software a známy PCI<sup>1</sup> hardware ktorý oplýva pokročilými možnosťami z meracej či kontrolnej sféry. V základe tento koncept ponúka chassis, kontrolnú jednotku a istý počet nahraditeľných

---

<sup>1</sup>Peripheral Component Interconnect

modulov. PXI zároveň predstavuje priemyselný štandard pod dohľadom PXI Systems Alliance (PXISA) [31]. Užívateľ teda môže ťažiť z toho, že využíva zaužívanú štandardizovanú technológiu, či už sa jedná o PXI ako platforme alebo PCI rozhraní. Ako bude platiť o väčšine produktov, takmer vždy bude existovať viac alternatív ako pre chassis, tak pre kontrolné jednotky, možno rátať s externými kontrolermi a mnohými riešeniami snažiacimi sa prispôbiť rôznym možným potrebám v priemysle. Práve na spomenutej kontrolnej jednotke bude bežať známy OS<sup>2</sup> (Win10, LabVIEW Real-Time, Win7-32b/64b), čo do vybavenia možno rátať napríklad s procesorovou Atom, Celeron, Xeon, Core i3-i7 Intel architektúrou a pamäťou ako v bežnom mainstreamovom PC, zatiaľ čo tento kontrolér zaberie len niekoľko slotov v PXI chassis. Odstavec vychádza zo spomínanej výročnej správy zdroja [30].

### 1.1.2 CompactRIO

Princíp ako funguje táto platforma je rovnaký ako v prípade PXI. Chassis CompactRIO už ale obsahujú vbudovaný kontrolér. K dispozícii sú aj CompactRIO Single-Board kontroléry, ale ako som už spomínal, National Instruments takmer všade ponúka istú alternatívu pre konkrétnu potrebu priemyslu. Zatiaľ čo moduly podporované platformou PXI mali rovnomenný prívlastok: PXI, vymeniteľné moduly podporované chassis CompactRIO sa nazývajú C-Series, kam spadá aj karta NI9219. Viac o CompactRIO systémoch sa možno dozvedieť zo zdroja [32], z kadiaľ zároveň vychádza tento odstavec.

### 1.1.3 PC-Based Systems

Inak aj nazývané Data Acquisition (DAQ) hardware and driver software, sa sústreďuje na zber analogových dát a ich transformáciu do digitálnej podoby, pre ďalšie spracovanie na PC. Jedná sa v základe o lacnejšiu alternatívu ako v prípade doteraz spomenutých riešení, lebo sa využíva výpočetnej sily tradičných výpočetných jednotiek v podobe napríklad klasických desktop počítačov. Platforma ponúka dve možnosti využitia a spojenia s PC. Jednou z nich je pripojenie multifunkčného I/O zariadenia priamo pomocou USB<sup>3</sup>, ethernetu, poprípade bezdrôtovej technológie, alebo pripojenie chassis CompactDAQ, na ktorú sa dá napojiť istý počet vymeniteľných C-Series modulov, znova ponúkajúc širokospektrálne výhodnú modularitu. PC-Based Systems sú ďalšou platformou vyhovujúcou pre NI9219, o tom si bližšie povieme v sekcii 1.4 Klasifikácia a použitie karty NI9219. Informácie v odstavci pochádzajú zo zdrojov [33], [30].

---

<sup>2</sup>Operating System - operačný systém

<sup>3</sup>Universal Serial Bus - Univerzálna sériová zbernica

## 1.2 Softwarové prostredia NI

Z preferencií užívateľa NI ponúka v oblasti SW aj iné alternatívy ako ich vlajkovú loď LabVIEW. Dvomi alternatívami sú Measurement Studio a NI LabWindows/CVI. [34]

### 1.2.1 Measurement Studio

NI Measurement Studio predstavuje nadstavbu prostredia Microsoft Visual Studio od spoločnosti Microsoft. To predstavuje jeden z možných dôvodov, prečo by sa mohol užívateľ rozhodnúť práve pre túto alternatívu, práca by sa mohla líšiť maximálne v používaní iných knižníc a built-in funkcií, ak užívateľ ovláda jeden z podporovaných jazykov, ktorými sú Visual C#, Visual Basic a .NET. Knižnice sú komplexného charakteru ponúkajúce možnosti generovania signálov rôznych vlastností ako aj poskytujú mnoho prostriedkov na analýzu dát a signálov pre abstrahovanie kritických informácií. Odstavec vychádza zo zdrojov [34] a [36].

Štúdio pozostáva z 3 hlavných častí, ktoré dôveryhodne kopírujú štruktúru na ktorej buduje LabVIEW. Obsahuje súbor ovládacích a zobrazovacích prvkov pre tvorbu aplikácií na user-interface báze, knižnicu rôznych analytických funkcií a sprievodcu pre spojenie s HW. Prvé zo spomenutých odpovedá vo sfére LabVIEW kontrolným a indikačným prvkom (Controls and Indicators), ktoré užívateľ pridáva na čelný panel (Front Panel) spôsobom drag-and-drop. Druhé spomenuté sú paralelou funkcií, ktoré v LabVIEW užívateľ pridáva na blokový diagram (Block Diagram). Sprievodca pre spojenie s HW za účelom jednoduchších meraní a ďalších funkcií predstavuje v LabVIEW sfére program NI MAX. V poslednej možnosti teda vie užívateľ v Measurement Studio generovať alebo čítať signál bez napísania čo i len čiarky kódu. Užívateľ prakticky používa Microsoft Visual Studio, pričom má v menu možnosť s názvom Measurement Studio, kde po rozkliknutí objaví Pomoc (Help), Príklady (Examples) a okrem iného aj možnosti pre nakonfigurovanie práce s prostredím. Pri tvorbe nového projektu ponúka štúdio rôzne šablóny v závislosti na potrebe danej vytváranej aplikácie. Užívateľ zadá detaily pre následovné meranie a Measurement Studio vytvorí základnú hierarchiu pre tvorbu následovného špecifického riešenia. Tým získa ideálne predpoklady pre rýchly proces vývoja chcenej aplikácie. Viac o Measurement Studiu sa nachádza v zdroji [36], z ktorého vychádza tento odstavec.

### 1.2.2 LabWindows/CVI

LabWindows/CVI predstavuje prostredie pre programovanie na báze a s použitím jazyka ANSI C. Okrem kompilátora jazyka C pod hlavičkou ANSI ale obsahuje

množstvo funkcií a knižníc pre prácu s NI HW. Zdrojový kód sa teda vie jednoducho budovať drag-and-drop systémom z databázy vbudovaných funkcií, samozrejme s možnosťou obohacovať túto databázu s vlastnými riešeniami, poprípade dostupnými online špecifickými prístupmi ostatných užívateľov. [34] [35]

Tak ako v LabVIEW, tak aj LabWindows/CVI ponúka možnosť tvorby špecifických užívateľských rozhraní (GUI - Graphical User Interface). Prostredie prakticky ponúka dve možnosti ako vytvárať GUI pre konkrétnu aplikačnú potrebu. Jednou je používanie volaní funkcií za pomoci ANSI C ako písaného programovacieho jazyka, druhou je používanie interaktívneho User Interface Editoru, ktorý obsahuje tlačítka, rozbaľovacie menu, grafy, schémy, prepínače a iné kontrolné či zobrazovacie mechanizmy, pomocou ktorých vie užívateľ vytvoriť rozsiahle GUI bez používania akéhokoľvek textového prístupu programovania v procese tvorby. [35]

## **1.3 LabVIEW**

### **1.3.1 Výhody využitia LabVIEW**

Viac ako 30 rokov je LabVIEW SW vlajkovou loďou firmy NI a spoločnosť sa k tomuto produktu sama vyjadruje, že je to ideálnym softwarovým riešením v oblasti merania, automatizácie a regulácie. Spolu s vyjadrením “The Software Is the Instrument” možno očakávať značne výraznú a dedikovanú pozornosť voči SW, ktorý vyvíjajú [37]. Výsledkom spomenutého je istá forma záruky, že LabVIEW je a bude preferovaným a podporovaným jazykom NI a ich produktov. [30]

Čo sa týka vývoja v oblasti automatizovaných testov, meraní a priemyselnom riadení, v roku 2015 vytvoril EE Times a EDN (magazíny z oblasti elektroniky) anketu [38] pod predmetom automatizovaných testov a pýtal sa, aký jazyk spytovaní inžinieri používajú. V ankete dominoval LabVIEW u 51% inžinierov z 345 opýtaných. Anketa bola „multiple response“, teda v tesnom závесе sa držal C/C++ so 47%, tretí Python s iba 25%, ktorý ale zaznamenal nadpriemerný vzrast. LabWindows/CVI na siedmom mieste so 14% a Measurement Studio na desiatom mieste s 5%. Spomenuté svedčí o relevancii jazyka medzi inžiniermi aj zo širokého spektra daného predmetu, nielen z dedikovaných kruhov NI, v ktorých sa v tejto práci zvyčajne pohybujeme.

### **1.3.2 Popis LabVIEW**

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) poskytuje grafické rozhranie pre vývoj a testovanie aplikácií v oblasti merania a regulácie, pričom náväznosť prostredia a jazyka na hardware je jeho príznačným prvkom. LabVIEW

je napísaný v C++ a .NET a jazyk v ktorom sa pomocou LabVIEW prvotne programuje, sa nazýva G language [41]. Ak teda užívateľ programuje pomocou LabVIEW, vytvára aplikácie, ktoré sa nazývajú VI (Virtual Instrument), voľným prekladom virtuálne prístroje. Kompatibilita LabVIEW kódu sa dá zaručiť takzvaným NI LabVIEW C Generátorom, ktorý je oficiálne prístupný z webov NI [39] a prepíše akékoľvek VI do ANSI C kódu.

Je dôležité spomenúť isté pojmy, ktorú sú významne spájané s LabVIEW. Využíva kompletne grafický programovací prístup, veľká časť programovania je na princípe drag-and-drop rôznych regulačných a zobrazovacích či funkčných prvkov. Sleduje tok dát (dataflow model), ktorý sa sústreďuje na zľahka povedané - splnenie podmienok pre všetky inputy, vtedy povolí operáciu daného blackboxu, na ktorý sú inputy napojené. Tradičný prístup, zvyčajne nazývaný aj Control Flow, v skratke naopak figuruje ako model zložený zo série operácií, ktoré sa vykonávajú v špecifickom poradí. Zdrojom informáci tohto úseku je dokument pod zdrojom [40], ktorý vie zároveň poslúžiť aj ako komplexnejší zdroj pre zorientovanie sa v oblastiach datovej akvizície pomocou LabVIEW.

Princíp a celá práca vrámci vývoja aplikácie v LabVIEW sa odohráva na dvoch obrazovkách, ktoré sa nazývajú blokový diagram (Block Diagram) a čelný panel (Front Panel). Čelný panel prakticky predstavuje GUI<sup>4</sup> koncovej aplikácie. Zoznam prvkov, ktoré užívateľ vkladá do čelného panelu, je v okne s názvom Controls (ovládacie prvky). Po pridaní objektu zo zoznamu ovládacích prvkov na čelnom paneli sa na blokovom diagrame zobrazí matematická bloková reprezentácia daného objektu. Takýmto spôsobom sa jednoducho povedané zobrazí na blokovom diagrame súbor prvkov, ktoré medzi sebou budú potrebovať istú logickú súvislosť, ktorú zabezpečíme vhodným prepojovaním prvkov a dopĺňovaním ich o prvky zo zoznamu funkcií (Functions). Zoznam „Functions“ slúži ako zdroj funkcií pre prácu v blokovom diagrame.

Pre pochopenie princípu VI, na ktorom je LabVIEW založený, možno človeku ználemu iných programovacích jazykov vysvetliť následovne: Tak ako sa môžeme v nízkoúrovňovom jazyku stretnúť so subroutine, poprípade v drvivej väčšine text-based jazykov sa môžeme stretnúť s funkciami, tú istú výhodu a princíp v LabVIEW zabezpečuje VI a vytváranie subVI vrámci iných VI a kľudne aj vrámci ďalších iných subVI atď. To je prístup ktorý pomáha väčšine programovacích jazykov udržiavať kód pochopiteľným a ľahko opakovateľným a LabVIEW nie je výnimkou. Tu nastáva

---

<sup>4</sup>Graphical User Interface - grafické užívateľské rozhranie



zlomový bod, kedy je dôležité aby si toho bol programátor vedomý a nevytvoril LabVIEW kód neuveriteľných rozmerov, ale používal princíp vnorených VI. V opačnom prípade je to problematické a ťažko pochopiteľné či udržiavateľné. V podstate ide o rovnaký princíp ako v klasickom negrafickom programovaní, no podľa môjho názoru je dodržanie tohto pravidla v grafickom programovaní o to podstatnejšie, pretože jednotlivé prvky poprepájané skrze celé VI navzájom sú ohromne zložité na trasovanie. Následné pochopenie istých súvislostí je potom o to viac problematickejšie, keďže sú jednotlivé objekty v skutočnosti prepojené reálne vyobrazenou grafickou čiarou a celé VI potom akurát budí dojem obrovskej kopy prepojujúcich datových tokov, pričom by ani nemuselo ísť o zložitý princíp, ktorý dané VI v skutočnosti vykonáva. [40]

## 1.4 Možnosti meracej karty NI 9219

### Klasifikácia a použitie karty NI9219

Meracia karta National Instruments 9219 má na každom kanále 24 bitový AD prevodník,<sup>5</sup> zariadenie je schopné súčasne vzorkovať všetky štyri analógové vstupné kanály. Zaraďuje sa do skupiny C series, čo je skupina modulov kompatibilných s chassis CompactDAQ a aj CompactRIO. Čo sa týka karty NI9219, je zariadenie kompatibilné so všetkými chassis modulmi od CompactRIO a to sú menovite CompactRIO FPGA,<sup>6</sup> CompactRIO Scan Mode, CompactRIO DAQmx a NI 9144/5 Scan Mode. Ďalej je kompatibilná so všetkými chassis modulmi CompactDAQ menovite samotné CompactDAQ, cDAQ-9172, WLS/ENET-9163 a USB-9162 [11], kde je ale nutné podotknúť, že posledné 3 spomenuté sú zastarané a čo sa týka podpory, tak nepodporujú mnoho iných kariet C-Series [13]. Posledný spomenutý chassis bude platformou s ktorou budeme operovať pri vytváraní LabVIEW aplikácie vrámci tejto práce čo sa týka spojenia s PC a teda prostredím LabVIEW.

Pre náš projekt by bolo zbytočné využívať chassis o viac slotoch ako jedna. Čím sa náš výber vrámci potrieb našej aplikácie zníži na 4 ďalšie možnosti a tými sú cDAQ-9171, cDAQ-9181, cDAQ-9191 a čiste teoreticky aj WSL/ENET-9163. Posledný spomenutý, je už viacmenej zastaraný a pre naše účely vývoja, by sme nevyužili jeho ethernetovú alebo bezdrôtovú funkciu [16]. Teoreticky lepšou možnosťou by mohla byť ľubovoľná z kariet cDAQ-9171, cDAQ-9181, cDAQ-9191. Posledné dve spomenuté akurát využívajú bezdrôtové a ethernetové rozhranie ktoré ale vyslovene nepotrebujeme, skôr sa to hodí prakticky „do terénu“ a pozorujeme pri nich aj

---

<sup>5</sup>Analog-to-Digital Converter - analogovo digitálny prevodník

<sup>6</sup>Field-Programmable Gate Array - Programovateľné hradlové pole

zvýšenie ceny pre funkciu, ktorú reálne nevyužijeme, či už prístup alebo zvýšená rýchlosť [15]. Vedeli by sme teda dôjsť k jednému hlavnému dôvodu, prečo je chassis modul cDAQ-9171 vhodnejší pre náš účel ako NI USB-9162. Karta NI USB-9162 [17] je staršia a software pracujúci s prepojením NI HW a PC nazývaný NI MAX bude rozoznávať chassis NI USB-9162 a cDAQ-9171 rozdielne, hoci by do nich bol zasunutý ten istý modul - napríklad naša karta NI 9219. To môže vytvoriť problém v situácii, že chcete jednoducho zameniť NI USB-9162 + NI 9219 za cDAQ-9171 + NI 9219. V takomto prípade je nutnosť zasiahnuť do kódu, no nie všade sa nachádza zdrojový súbor a v priemysle je možné, že budú v istých situáciách existovať len „executables“ - spustiteľné súbory. Situácia teda môže vo výsledku vyeskalovať do problémov s kompatibilitou. [14]

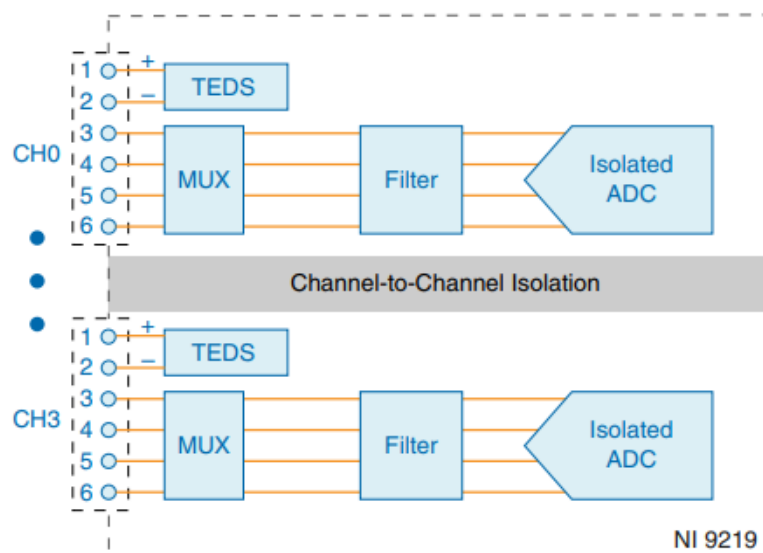
Keďže cDAQ má aj viacslotové alternatívy a ponúka celú hromadu chassis, bolo by v rámci budúcnosti ideálne využiť chassis cDAQ-9171, na ktorom by sa vytvoril istotne kompatibilný kód so všetkými takýmito konštaláciami rôznych modulov CompactDAQ systémov.

Ďalším možným rozšírením HW už nevykonávajúcim funkčnú zmenu čo do kompatibility v rámci programovaného SW predstavujú extra konektory NI 9973 a NI 9972 ako backshell pre prvý spomínaný, čo znamená, že sa NI 9973 [20] napojí na NI 9972 [19], ktorý je napojený na ľubovoľný počet zo 4 kanálov karty NI 9219 [10]. Poskytujú nadštandardnú ochranu čo sa týka prenosu signálu a ochrany používaného HW ako aj ochranu pred vyšším napätím ktorému by mohol byť používateľ vystavený. Riešenie poskytuje robustnejšiu možnosť zavedenia karty NI 9219 v konkrétnej aplikácii. [18]

Zemnenie zariadenia a každý kanál sú navzájom elektricky izolované. Kanály sú individuálne voliteľné, z čoho vyplýva, že užívateľ môže vykonávať na každom zo štyroch kanálov iné meranie používajúc napríklad iný snímač. Pre isté módy je možnosť využiť prítomnosť obvodu budiaceho napätia, pre meranie signálov napríklad z odporových snímačov teploty alebo termoelektrických článkov (konkrétne hodnoty na konkrétne módy v datasheete [1] od str.12). Pravidlá rozloženia pinoutu pre prácu na konkrétnom meraní je konfigurované zariadením NI 9219, pre každý mód zvlášť. Vid. Tab.1.1.

## **AD prevodník**

AD prevodník karty NI 9219 pracuje na princípe delta-sigma modulácie, čo v skratke znižuje nároky na rozlíšenie prevodníka na úkor následne potrebnej zvýšenej vzorkovacej frekvencie. Zároveň znižuje požiadavky pre náročnosť analogových obvodov na úkor digitálnych. [22]



Obr. 1.1: Základné obvodové schéma:[1]

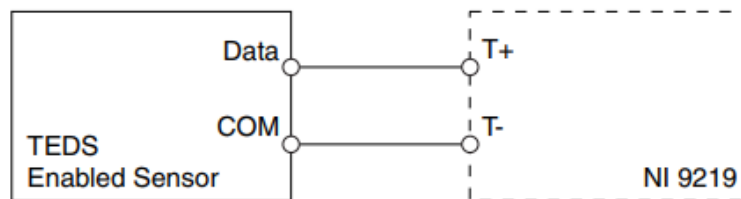
Zľahka si predstavme proces prenosu signálu kartou NI 9219. Na kanály a jednotlivé svorky zariadenia prichádza analogový - spojitý signál, ktorý za účelom spracovania potrebujeme v digitálnej - nespojitej podobe. V reálnom svete možno očakávať signály s rozpínajúcim sa spektrom frekvencií, čo nám zabraňuje určiť vhodnú vzorkovaciu frekvenciu, čo rieši vbudovaný analogový anti-aliasingový dolnopriepustný prefilter. Po filtrácii spomínaným filtrom sa signál vzorkuje v súlade s Nyquistovým vzorkovacím teorémom, čo predstavuje frekvenciu minimálne 2 krát vyššiu čo do hodnoty v porovnaní s medznou frekvenciou nášho dolnopriepustného filtru. Karta NI9219 poskytuje rýchlosť odoberania vzoriek o hodnote 100 S/s/channel, čo predstavuje 100 vzoriek za sekundu na každý kanál. Karta odoberá vzorky na kanáloch simultánne [49]. Signál sa teda vzorkuje bez obáv výskytu aliasingu AD prevodníkom a každá vzorka je kvantifikovaná v našom prípade 24 bitmi, vo všeobecnom prípade to bude vždy konečný počet bitov, ktorý je daný prevodníkom. [1], [21]

## TEDS

Zapojenie PIN1 = T+ ; PIN2 = T-

Pre ľubovoľný typ merania sa zapojenie PINov nemení, viď. Tab.1.1. TEDS (Transducer Electronic Data Sheet) sa dá zľahka preložiť ako elektronický datasheet určitého senzoru. Je to datový súbor ktorý sa nachádza priamo v analogovom senzore (zvyčajne pamäť EEPROM<sup>7</sup>), prípadne ak ide o zastaraný senzor, poprípade senzor bez internej pamäte, môže byť voľne dostupný napríklad na webe pod

<sup>7</sup>Electrically Erasable Programmable Read-Only Memory - elektronicky mazateľná pamäť, programovateľná na využitie iba na čítanie



Obr. 1.2: TEDS zapojenie [2]

názvom Virtual TEDS. Štandard IEEE<sup>8</sup> 1451.4 definuje štruktúru TEDS súboru. Táto štruktúra má sekcie, ktoré pevne daným spôsobom identifikujú detaily používaného zariadenia. „Basic TEDS“, ako prvá sekcia obsahuje informácie pre elementárnu identifikáciu zariadenia. Obsahuje napríklad meno výrobcu, typ senzoru, číslo modelu alebo sériové číslo. [8] Následne „IEEE standard TEDS“ popisuje dôležité vlastnosti daného senzoru. Nie nevyhnutne môže nasledovať šablóna „Calibration TEDS“, ktorá podľa očakávaní z princípu kalibrácie zariadenia obsahuje informácie o vstupno-výstupnej krivke senzoru. Pre vhodnú predstavu, môže ísť o páry hodnôt reprezentujúce input vs. output vo forme kalibračnej tabuľky. Čo sa týka SW možností získania TEDS informácií senzoru z oblasti National Instruments, prostredníctvom NI MAX sa ľahko importujú kalibračné údaje z práve používaného senzoru, poprípade je aj pri TEDS senzoroch dostupná Virtual TEDS online alternatíva. [9]

Ako z odznených informácií mohlo vyplývať, tento štandard ponúka priemyslu rýchlejšie zavedenie a použitie senzoru, keďže systém používajúci senzor s TEDS má prístupných mnoho v tomto smere kritických informácií. Systém senzor identifikuje, zistí typ/model a konkrétne vlastnosti z toho vyplývajúce. Určí sa rozhranie v ktorom operuje, čím v dôsledku týchto faktov systém spozná ako správne získavať signál z analogového senzoru. Fakt, že sa to stáva štandardom tomu navyše dodáva kredibilitu a istotu rámci kompatibility v budúcnosti. [13], [7]

### Pinout vysvetlivky:

EX+ kladný pin pre budiaci obvod

EX- záporný pin pre budiaci obvod

HI kladný vstupný signál

LO záporný vstupný signál

T+ TEDS data

T- TEDS com

Tento zoznam bol čerpaný z [2].

<sup>8</sup>Institute of Electrical and Electronics Engineers - Inštitút pre elektrotechnické a elektronické inžinierstvo

Tab. 1.1: Zhrnutie zapojenia signálov pre ľubovoľný kanál pre všetky módy [2]

Mód	PIN1	PIN2	PIN3	PIN4	PIN5	PIN6
Voltage	T+	T-	-	HI	LO	-
Current	T+	T-	HI	-	LO	-
4-Wire Resistance	T+	T-	EX+	HI	EX-	LO
2-Wire Resistance	T+	T-	HI	-	LO	-
Thermocouple	T+	T-	-	HI	LO	-
4-Wire RTD	T+	T-	EX+	HI	EX-	LO
3-Wire RTD	T+	T-	EX+	-	EX-	LO
Quarter-Bridge	T+	T-	HI	-	LO	-
Half-Bridge	T+	T-	EX+	HI	EX-	-
Full-Bridge	T+	T-	EX+	HI	EX-	LO
DI	T+	T-	-	HI	LO	-
Open Contact	T+	T-	HI	-	LO	-

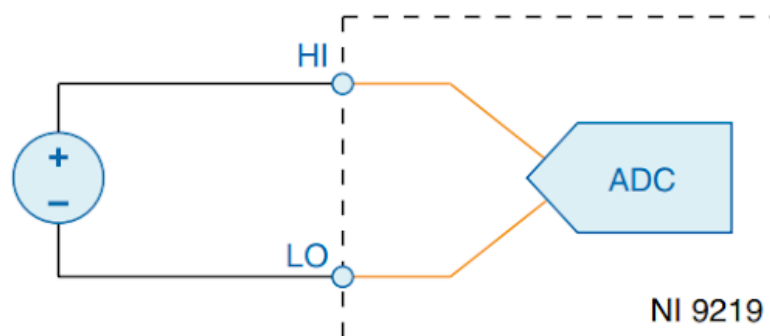
### Typy meraní pod kartou NI 9219 [2]

1. Napätie (Voltage)
2. Prúd (Current)
3. 4-vodičový odpor (4-Wire Resistance)
4. 2-vodičový odpor (2-Wire Resistance)
5. Termoelektrický článok (Thermocouple)
6. 4-vodičové meranie odporovým teplomerom (4-Wire RTD)
7. 3-vodičové meranie odporovým teplomerom (3-Wire RTD)
8. Štvrtinový mostík (Quarter-Bridge)
9. Polovičný mostík (Half-Bridge)
10. Plný mostík (Full-Bridge)
11. Digital In \*\*
12. Open Contact \*\*

\*\* podporované v systémoch CompactRIO

#### 1.4.1 Napätie (Voltage)

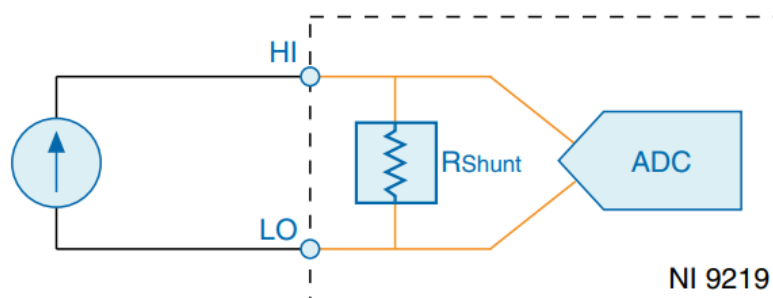
Pre meranie napätia na meracej karte NI 9219 viď. Obr.1.3 pripojíme zdroj signálu na PINy 4 pre HI terminál a 5 pre LO terminál, následne AD prevodník zaznamenáva hodnotu napätia medzi použitými PINmi. [6] Rozsahy napätí ktoré poskytuje karta NI 9219 sú  $\pm 60$  V,  $\pm 15$  V,  $\pm 4$  V,  $\pm 1$  V,  $\pm 125$  mV. [1]



Obr. 1.3: Schéma zobrazujúca zapojenie karty NI 9219 pre meranie napätia [1]

### 1.4.2 Prúd (Current)

Zariadenie NI 9219 meria hodnotu prúdu Obr.1.4 pomocou merania napätia na odporovom bočníku označenom  $R_{Shunt}$ . Časť obvodu, ktorá obsahuje AD prevodník bude mať vysoký vstupný odpor, teda očakávame, že všetok prúd bude prechádzať rezistorom o známej hodnote. Následne budeme merať napätie na rezistore  $R_{Shunt}$  z čoho sa pomocou Ohmova zákona dopočíta prúd prechádzajúci svorkami. Rozsah možného prúdu na vstupe sa rovná  $\pm 25$  mA. [1]



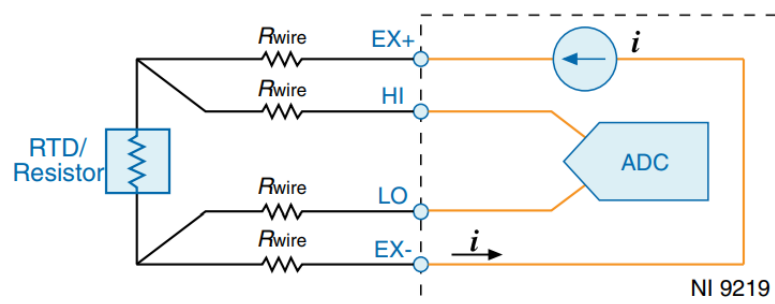
Obr. 1.4: Schéma zapojenia kanálu karty NI 9219 pre meranie prúdu [6]

### 1.4.3 4-Vodičové meranie odporu (4-wire resistance) a Odporový teplomer RTD v 4-vodičovom zapojení

Vid. Obr.1.5. Zariadenie NI 9219 vnesie do obvodu prúd o známej veľkosti, pričom časť obvodu obsahujúca AD prevodník pohltí len zanedbateľný objem prúdu v dôsledku vysokej impedancie. Veľkosť vneseného prúdu závisí na veľkosti odporu záťaže. Hodnota odporu sa vypočíta z aplikovania Ohmovho zákona pri znalosti

hodnoty napätia z merania na záťaži RTD/Resistor. Rozsahy pre 4-vodičové meranie odporu sú 10 k $\Omega$  a 1 k $\Omega$ , rozsahy pre odporový teplomer (RTD) v 4-vodičovom zapojení sú 5.05 k $\Omega$  (Pt 1000) a 505  $\Omega$  (Pt 100). [1]

RTD (Resistance Temperature Detector) využíva už spomenutý prechod konštantného prúdu RTD súčiastkou. Zistením napätia z merania vypočítame hodnotu odporu, ktorá priamo závisí na teplote materiálu RTD súčiastky. Zvyčajne sa používa navinutý kábel o malom priemere, ktorý bude príznačný vysokou čistotou daného materiálu - najčastejšie platina, poprípade nikel alebo meď. Použitím platiny sa docieli dlhotrvámejšia stabilita pri meraní, vyšší teplotný rozsah, presnejšie meranie. Platina je mimo iného aj odolnejšia voči oxidácii. [23]



Obr. 1.5: Schéma zapojenia kanálu karty NI 9219 pre 4-vodičové meranie odporu a odporový teplomer v 4-vodičovom zapojení [1]

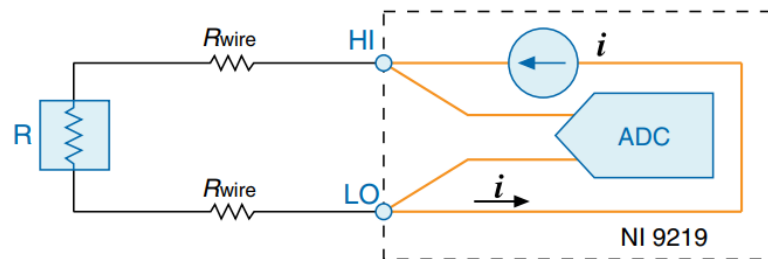
#### 1.4.4 2-Vodičové meranie odporu (2-Wire resistance) a Štvrtinový mostík (Quarter-bridge)

Vid. Obr.1.6. Princíp analogicky zopakovaný ako v prípade 4-vodičového usporiadania sekcie 1.4.3, nevyhneme sa však odporu prívodných vodičov  $R_{wire}$ . Rozsahy pre štvrtinový mostík sú 350  $\Omega$  a 120  $\Omega$ , rozsahy pre 2-vodičové meranie odporu sú 10 k $\Omega$  a 1 k $\Omega$ , čo sa zhoduje so 4-vodičovým usporiadaním. [1]

#### 1.4.5 Termoelektrický článok (Thermocouple)

Vid. Obr.1.8. Termoelektrický článok je založený na prevode tepelnej energie na elektrickú. Pozorovať ho môžeme pri spojení v podstate akýchkoľvek dvoch čo do termoelektrických vlastností - rôznych vodičov. V rámci určenia konkrétnej teploty, každý takýto pár generuje známu hodnotu napätia, ktorá je matematickým vzťahom spojená teplotnou hodnotou.

Čo sa týka reálnej implementácie, stretávame sa so štandardizovanými párami materiálov pre účely termoelektrických článkov. Tieto dvojice sa označujú jedným veľkým



Obr. 1.6: Schéma zapojenia kanálu karty NI 9219 pre 2-vodičové meranie odporu a štvrtinový mostík [1]

písmenom. V USA je ich štandardizovaných napríklad 8: B, E, J, K, N, R, S, T [23]. Vo všeobecnosti ale isté krajiny alebo komisie podporujú niektoré dvojice navyše. Celkovo má každý typ ďalej aj farebné označenia pre jednotlivé nohy článku, tie sa líšia či už od krajiny alebo komisie, takže jednoznačne oblasť veľmi volatilná a rozhoduje či sledujeme štandardy IEC<sup>9</sup>, poprípade nakupujeme z Anglicka, či USA, kde všade možno očakávať rôzny štandard ako aj označenie.

Jav môžeme pozorovať v uzavretom alebo otvorenom obvode. V uzavretom obvode bude prechádzať prúd za podmienky, že by mali spoje rozdielnu teplotu. Čo by za splnenia podmienky rôznych materiálov čo do termoelektrických vlastností malo stačiť, každopádne vynímajúc teploty veľmi nízkych hodnôt pri istých druhoch materiálov. Mnohé používané materiály (páry materiálov pod konkrétnym typom termočlánku) značne poklesnú na termoelektrickej citlivosti už pod bodom mrazu. Seebeckov koeficient všetkých vodičov je zanedbateľný pri 0 K. V druhom prípade, teda pri otvorenom obvode sa objaví na vzniknutých svorkách elektrické napätie. Celý princíp termoelektrických článkov je postavený na Seebeckovom jave a aj spomenuté vzniknuté napätie je dané Seebeckovými vzťahmi. [24]

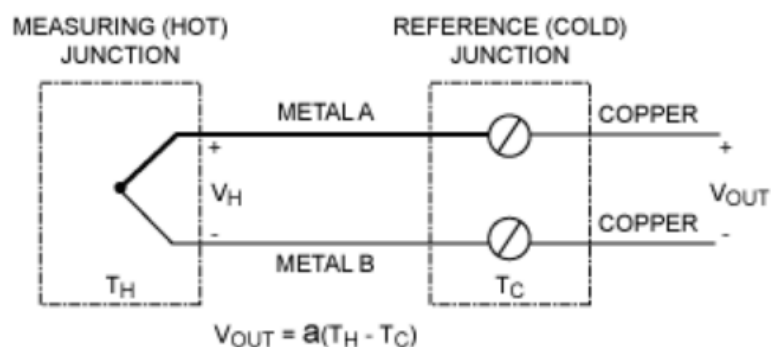
V karte NI 9219 sa spravidla používa rozsah  $\pm 125$  mV AD prevodníka pre vrátenie hodnoty meraného napätia. Použitie tohto módu sa odporúča zahajovať za stálych teplotných podmienok v okolí modulu alebo pripojovacích konektorov. Každý kanál má vbudovaný termistor<sup>10</sup> pre CJC - Cold Junction Compensation [1]. Kompenzácia studeného konca (CJC) je schematicky popísaná na Obr.1.7 a informácie témy CJC som z teoretického hľadiska čerpal zo zdrojov [26] a [25], kde sa možno dostať ku ďalším informáciám. Zo vzťahov z Obr.1.7 je jasné, že  $V_{OUT}$  je funkcia rozdielu horúceho (meraného) spoja a chladného (referenčného) spoja. Ďalej vidíme, že sa ako sprostredkujúci kov využila meď, pričom za stálosti rovnakej teploty oboch

<sup>9</sup>International Electrotechnical Commission

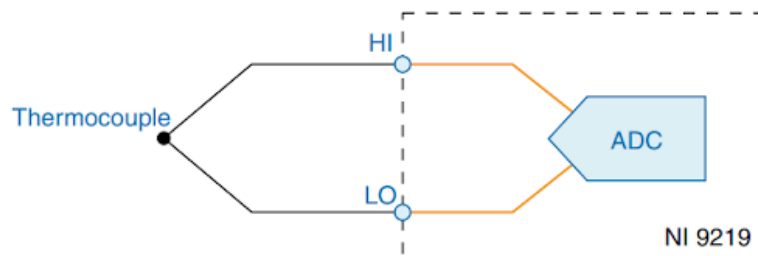
<sup>10</sup>Teplotne citlivý rezistor



spojov v chladnom (referenčnom) spoji na materiále tohto kovu nezáleží a na meranie nebude mať vplyv. Je ale nutné hodnotu tejto teploty poznať. Zvyčajne sa používa (kedysi normovaných 0°C), každopádne, pre praktické dôvody tomu tak vždy byť nemusí, následne sa táto nenulová referenčná hodnota na chladnom spoji musí dodatočne kompenzovať na výstupnom napätí. Na to sa používa už spomenutý vbudovaný termistor. Termistory v takýchto aplikáciach majú výhodu vo veľkom rozsahu, bohužiaľ na úkor linearity. Vhodnejšími alternatívami sú rôzne integrované obvody na kremíkovej báze snímania teploty, poprípade platinové odporové snímače teploty kde ale prvý spomínaný pracuje v užšom rozsahu a druhé spomínané riešenie je finančne náročnejšie. [23]



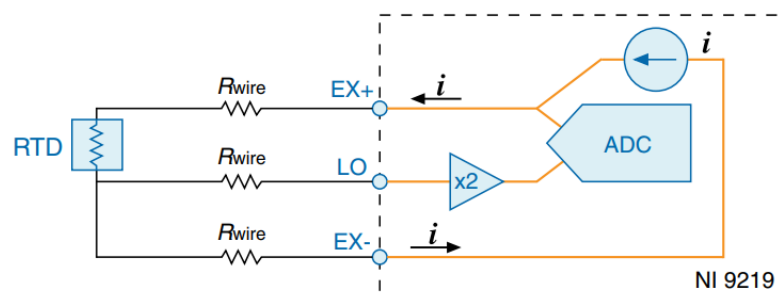
Obr. 1.7: Kompenzácia studeného konca [26]



Obr. 1.8: Schéma zapojenia kanálu karty NI 9219 pre termoelektrický článok [1]

#### 1.4.6 Odporový teplomer (RTD) v 3-vodičovom zapojení

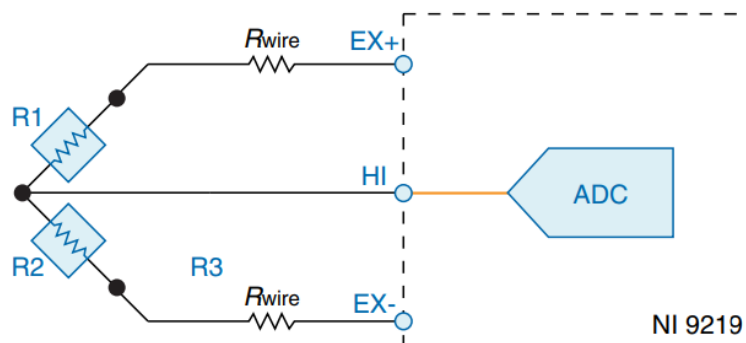
Medzi terminálmi EX+ a EX- je budený prúd zo zariadenia NI 9219. Zariadenie v oblasti HW kompenzuje odpor prírodných vodičov aplikovaním 2-násobného zosílenia napätia na zápornom prívode do AD prevodníka rezultujúc vo vykompenzovanie účinku odporu prírodných vodičov na kladnom prívode AD prevodníka. Schéma na Obr.1.9, vstupné rozsahy sú totožné so 4-vodičovým usporiadaním. [1]



Obr. 1.9: Schéma zapojenia kanálu karty NI 9219 pre odporový teplomer v 3-vodičovom zapojení [1]

### 1.4.7 Polovičný mostík (Half-bridge)

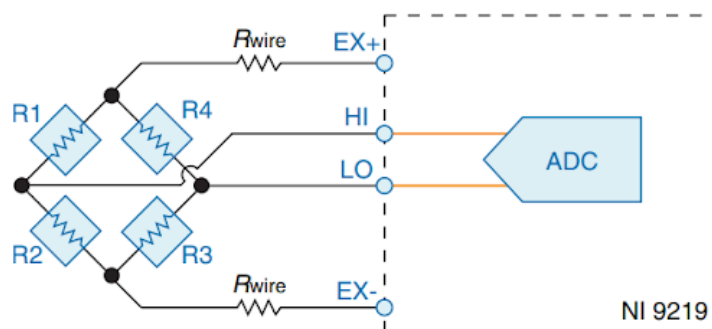
Vid. Obr.1.10. Vstupný rozsah pre polovičný mostík činí  $\pm 500$  mV/V. [1] Mostíkové zapojenia všetkých druhov sa používajú na meranie malých zmien odporu. Ideálny prípad pri použití tejto metódy merania je v plnom mostíku zapojenie, kedy v našom prípade na Obr.1.11 sa budú odpory R1 a R3 meniť jedným smerom (teda rásť/klesať) a odpory R2 a R4 meniť smerom opačným. Takto zaručíme čo najvyššiu citlivosť mostíka, čo je jeho primárnym účelom. Veľmi častou aplikáciou sú práve tenzometre, súčiastky, ktorým sa mení odpor v závislosti na ich mechanickej deštrukcii - napríklad predĺženie, alebo naopak skrátenie, čo v následku alteruje ich odpor. [27], [28]



Obr. 1.10: Schéma zapojenia kanálu karty NI 9219 pre polovičný mostík [1]

### 1.4.8 Plný mostík (Full-bridge)

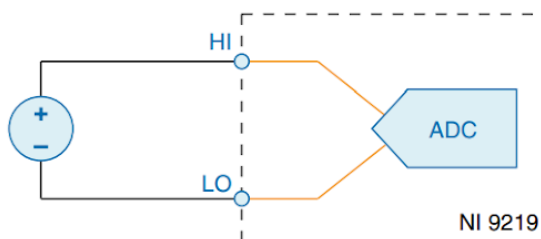
Vid. Obr.1.11. Popísaný je v rámci polovičného mostíku sekcie 1.4.7. Vstupné rozsahy činia  $\pm 62.5$  mV/V a  $\pm 7.8$  mV/V. [1]



Obr. 1.11: Schéma zapojenia kanálu karty NI 9219 pre plný mostík [1]

### 1.4.9 Digital In

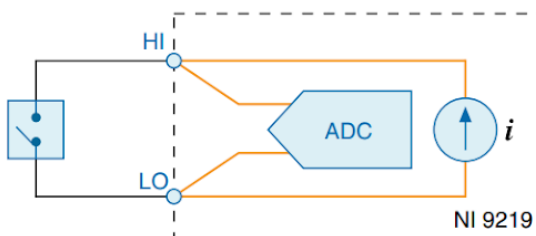
Vid. Obr.1.12. Napätový práh až do 60V (1,5 V implicitne). Funkcia je podporovaná len s použitím chassis CompactRio. [1], [29]



Obr. 1.12: Schéma zapojenia kanálu karty NI 9219 pre Digital In [1]

### 1.4.10 Open Contact

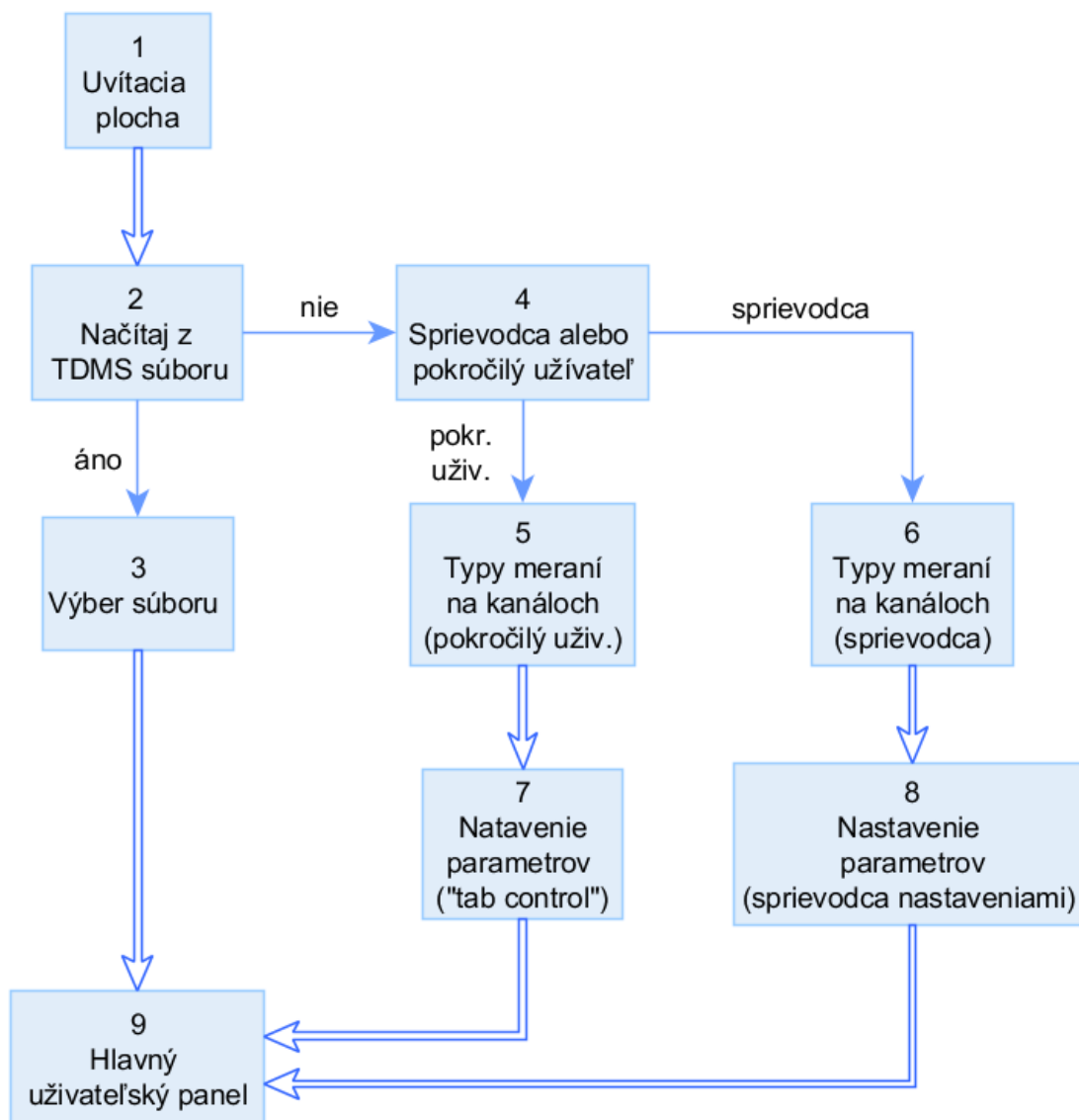
Vid. Obr.1.13. Zariadenie poskytuje prúd medzi HI a LO, následne zaznamenáva, či sú spomínané terminály „otvorené“ alebo „uzavrené“ meraním prúdu nimi prechádzajúcim. V prípade otvoreného obvodu je treba zaistiť aby nebolo vpúšťaných viac ako  $\pm 60$  V, vstupný rozsah je zadávaný ako 1,05 k $\Omega$ . Funkcia je podporovaná rovnako ako v prípade Digital In len s použitím chassis CompactRio. [1]



Obr. 1.13: Schéma zapojenia kanálu karty NI 9219 pre Open Contact [1]

## 2 Návrh zhotovenia praktickej časti

Nasledovný text posluží na opis našich plánov pre vytvorenie aplikácie. V podkapitolách bude popísané plánované užívateľské rozhranie (UI - User Interface) a možná implementácia povinných súčastí našej aplikácie ako napríklad spôsob riešenia pre ukladanie a načítanie parametrov, asynchrónna či sekvenčná architektúra pre logovanie a akvizíciu dát alebo spravovanie užívateľských inputov pomocou pollingu alebo Event štruktúry.



Obr. 2.1: Blokové schéma pre plánované UI

## 2.1 Návrh a popis UI

Na uvítacej ploche užívateľ aplikácie uvidí názov programu: Mnemosyne a typ meracej karty s ktorou sa pracuje, čo predstavuje kartu NI 9219. Po menšom časovom oneskorení v merítke sekúnd sa aplikácia presunie do stavu možnosťného výberu už existujúceho TDMS súboru, čo prebieha v krokoch 2 a 3. Ak si užívateľ vyberie nahrať súbor v ktorom boli logované dáta z akéhokoľvek predchádzajúceho merania, informácie o konkrétnom meraní sa zapisujú do všetkých potrebných premenných. Konkrétne hovoríme o špecifickom type merania na každom zo 4 možných kanálov a o hodnotách všetkých parametrov, ktoré dané typy merania potrebujú (hodnota excitačného prúdu alebo napätia, jednotka pre meranú veličinu, rozsah snímanej veličiny atď.). Následne by po načítaní všetkých týchto dát mohol užívateľ zopakovať meranie na základe predchádzajúcich nastavení, pričom na výber destinácie novo-vytvoreného súboru vyskočí okno alebo bude jeho trasa implementovaná v nastaveniach.

Ak užívateľ nechce načítať parametre z TDMS súboru, rozhodne sa, či chce sprievodcu nastaveniami, alebo najjednoduchšiu cestu pre zadanie merania a jeho parametrov v podobe pokročilého používateľa.

### 2.1.1 Pokročilý používateľ

Po výbere tejto možnosti sa užívateľovi zobrazia 2 okná, kým prejde do stavu „Hlavný užívateľský panel“. Tieto okná reprezentujú stavy 5 a 7 na Obr.č.2.1. Prvé okno predstavuje výber kanálu, pričom si na každom zo štyroch kanálov vyberie typ merania ktorý na ňom bude chcieť merať, bez akýchkoľvek pridaných funkcií. Následne sa po potvrdení výberu zobrazí okno s nastaveniami, ktoré bude pozostávať hlavne z jedného obsažného Control tabu, ktorý bude mať celkovo 4 karty, každá reprezentujúca jeden kanál a daný typ merania naväzujúci na kanál. Po rozkliknutí každej karty sa zobrazia všetky parametre pre daný kanál a typ merania jemu prisluchujúci, rovnako ako pri voľbe kanálu - bez akýchkoľvek funkcií a vysvetliviek navyše.

### 2.1.2 Sprievodca

Jeho funkciou bude užívateľovi poskytnúť vysvetlenia na každú jednu možnosť čo sa týka ako výberu typu merania, tak následne pre nastavenie hodnôt parametrov. Prechádzajúc cez nastavenia typu merania sa pri výbere jednotlivých možností zobrazia nápovedy, či už v podobe „Tip Strip“, alebo Description, či Visible Label/Free Label. Nápoveda užívateľovi priblíži o aký typ merania ide, na čo a za akých podmienok sa dá použiť.

V nastaveniach parametrov proces prebehne na rovnakom princípe. Aplikácia ponúkne v prvom rade užívateľovi prednastavené hodnoty, ktoré meranie rozbehnú bez jeho pričinenia. Spomenutým spôsobom ale taktiež poskytuje vysvetlenia jednotlivých parametrov. V prípade typu merania termočlánkom (Thermocouple), by aplikácia priblížila užívateľovi o aký typ termočlánku (Thermocouple type) sa jedná z hľadiska farebného označenia - v prípade používania neznámeho materiálu obsahujúceho farebné označenie. Aplikácia by mohla zároveň poskytovať jednoduché vysvetlenie kompenzácie studeného konca (CJC), čím by ku príkladu vysvitla funkcia nastavení „cjc source“, „cjc value“ alebo „cjc channel“, ktoré predstavujú zdroj, hodnotu a kanál kompenzácie studeného konca.

### 2.1.3 Hlavný užívateľský panel

Načítaním merania z TDMS súboru, prechodom nastaveniami Sprievodcom alebo pomocou „Pokročilého užívateľa“, sa užívateľ ocitne v Hlavnom užívateľskom paneli. Tu bude mať k dispozícii prostriedky na štart merania, sledovanie jednotlivých priebehov v grafe, zastavenie merania a bezpečné ukončenie aplikácie.

#### Tlačidlo Settings

Je na zváženie implementácia tlačidla „Settings“, ktoré by otvorilo ponuku nastavenia parametrov pre všetky vybrané typy merania. UI tohto okna by bolo zhodné s oknom ktoré sa poskytne pokročilému užívateľovi v bode 7 na Obr.č.2.1. Užívateľ by mohol dodatočne zmeniť nastavenia či už sa do Hlavného užívateľského panelu dostal akoukoľvek cestou. Obzvlášť užitočné by to bolo v prípade načítania súboru z TDMS, kedy by chcel používateľ urobiť minoritné zmeny v parametroch a nechcel by strácať čas prakticky prepisovaním všetkých parametrov klasickou dvojkrokovou cestou či už cez Sprievodcu alebo Pokročilého užívateľa.

### 2.1.4 Zmeny okien

Zmeny okien po výbere danej preferencie môžu nastať uzatvorením a otvorením nového dialógového okna, poprípade čelného panelu, alebo sa môžu jednotlivé okná reprezentujúce stavy z Obr.č.2.1 zobrazovať ako časti jedného obsažného čelného panelu, ktorého súčasťou sú všetky možné nastavenia a kontrolné prvky s ktorými by sme potrebovali pracovať. Následne sa pomocou Property nodes posúvame oknom vyhradenou veľkosťou po jednom obsažnom čelnom paneli.

Výhody a nevýhody spomenutých dvoch spôsobov spočívajú v znovupoužitelnosti a

rýchlosti. V oboch vyhráva možnosť použitia nových okien so separátnymi čelnými panelmi. Veľký čelný panel je ťažký na obsluhu pre developera v prípade zložitých veľkých projektov, pomalý na reakciu pri tvorení kódu, neprehľadný a len z ťažka znovupoužiteľný pre iné účely. Každopádne pre potreby našej aplikácie je otázne pre ktorú možnosť sa rozhodnúť, keďže bude druhá spomenutá, menej vyhovujúca možnosť jednoduchšia na implementáciu a čo do jej veľkosti nebudú spomínané negatíva až tak citelné.

## **2.2 Návrh a popis funkčnej časti programu**

Najrýchlejšiu a najjednoduchšiu verziu funkčného programu s daným zadáním si možno predstaviť pod stavovým automatom, kde je používateľ vždy v istom stave a do ďalšieho sa dostane po jednoduchom časovom oneskorení, alebo o ďalšom stave rozhodne sám užívateľ aplikácie. Vzhľadom na tento fakt možno očakávať využitie Event štruktúry, poprípade rozhodnutia na jej báze, teda konkrétny kontakt užívateľa s jemu poskytnutým UI vyvolá zmenu stavu. Po úspešnom splnení prerekvízit pre štart merania užívateľ zotrvá v akvizičnej slučke, kde prebehne meranie na základe požiadavok, ktoré užívateľ definoval prechodom stavovým automatom, kde možno očakávať výskyt všetkých potrebných nastavení.

### **2.2.1 Riešenie clusteru so všetkými nastaveniami**

Cluster, obsahujúci Controls, s ktorými budeme môcť nastaviť ktorýkoľvek typ merania, musí obsahovať všetky potrebné datové typy, spolu s množstvom ich maximálneho výskytu naprieč všetkými typmi merania. V tomto bode je potreba si prejsť všetky typy merania podporované kartou NI9219 a zistiť z čoho sa tento cluster bude skladať, bez zbytočnej duplikácie akéhokoľvek typu.

Ak sa dopracujeme k tomuto clusteru, je treba dbať na fakt, že na každom kanále sa môže merať iná veličina s inými nastaveniami a teda potrebuje svoju vlastnú inštanciu. Tento problém možno riešiť štyrmi kópiami tohto clusteru, vychádzajúc z definície pôvodného. Každá by bola fyzickou samostatnou kópiou, ktorá by mala vyhradenú vlastnú pamäť a prislúchala by k jednému zo štyroch kanálov karty.

Ak by sme chceli elegantnejšie riešenie, je možné zo spomínaného clusteru urobiť pole (Array) a inicializáciou jeho prvkov sa rovnako vytvoria pamäťové stopy ako v prípade fyzických kópií spomenutých v predchádzajúcom odstavci. Pomocou funkcie Index Array by potom bolo možné pristupovať k prvkom tohto poľa ako k prakticky samostatne vytvoreným clusterom.

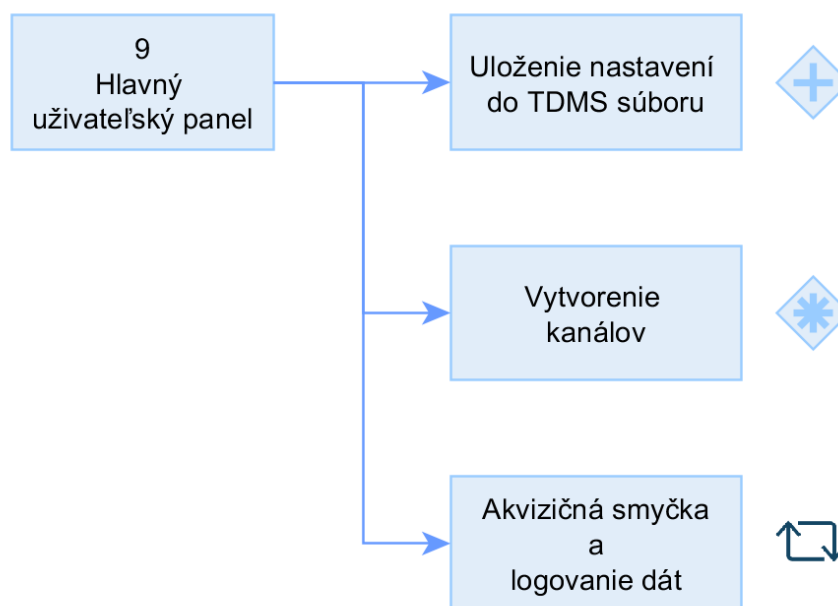
## 2.2.2 Sekvenčná časť vo forme stavového automatu

Uvítacia plocha, bod 1 v Obr.č.2.1, by mohla predstavovať prvý stav v stavovom automate, ktorý s časovým opozdením prejde do stavu, kde sa má užívateľ rozhodnúť o načítaní súboru z TDMS. V uvítacej ploche by sa mohlo nainicializovať pole clusterov, poprípade by sa mohla stanoviť hodnota všetkých využívaných Controls vrámci UI, ako aj ich Property node vlastností ako Disabled, Visible atď.

Načítanie súboru z TDMS bude naväzovať na spôsob akým parametre do TDMS súboru zapíšeme. Podstatou týchto častí programu budú funkcie TDMS Set/Get Properties. Ku získavaniu parametrov zapísaných v hlavičke TDMS by sa dali využiť výstupy tejto funkcie s názvami „property names“ a „property values“, ktoré by vedeli vypísať všetky názvy parametrov v hlavičke spolu s korešpondujúcimi hodnotami. Tým pádom by sa na základe týchto údajov určilo, ktoré parametre v danom clusteri využijeme a hodnoty by sa zapísali do príslušných Controls. Druhá možnosť predstavuje vyčítanie jednotlivých parametrov po mene a získavanie hodnoty parametru s daným menom, ktoré poskytneme funkcii TDMS Get Properties vo forme reťazca (String).

Ak užívateľ vyberie možnosť načítania z TDMS súboru, výber súboru prebehne tradičnou cestou otvorenia dialógového okna, kde užívateľ vyhľadá daný potrebný súbor a aplikácia sa po výbere ocitne v akvizičnej slučke ktorú popíšem na záver, po opísaní ostatných ciest ktorými sa k nej dá dostať. Bod 4 na Obr.č.2.1 sa v stavovom automate implementuje ako samostatný stav kde budú tlačidlá, ktorých stlačenie ukončí prebiehajúcu slučku s rozdielnymi hodnotami pre ďalší stav. Typ merania bodu 5 alebo 6 sa najľahšie implementuje ako výčet (enum). Pričom každý kanál bude reprezentovať jeden výčet, na ktorom užívateľ určí typ merania. Možnostne by to mohlo byť veľké pole prepínačov (switch), ktoré ponesú názvy konkrétnych typov merania. Užívateľ by iba 4 krát (počet kanálov NI9219) klikol na typ, ktorý si praje a danou udalosťou sa nastaví korešpondujúce hodnoty pre typ merania pre ten daný kanál. Na základe výberu v tomto kroku sa v krokoch 7 a 8 užívateľovi zobrazia len relevantné parametre pre nastavenie. Sprístupnenie špecifických Controls sa bude ovládať pomocou prvkov pre ovládanie vlastností (Property nodes) a manipulácia nastane možnosťne s boolean vlastnosťou ovládajúcou viditeľnosť prvku (Visible) a výčtom kontrolovanou vlastnosťou pre ovládanie možnosti manipulácie s prvkom (Disabled).





Obr. 2.2: Funkčné rozdelenie bodu 9 z Obr.č.2.1

### 2.2.3 Akvizícia dát

Akvizícia a logovanie dát sa uskutoční vrámci bodu 9, z Obr.č.2.1. Či už vyčítaním z TDMS súboru alebo nastavením pomocou „Sprievodcu“ alebo „Pokročilým užívateľom“ má aplikácia hodnoty všetkých potrebných parametrov, ktorými sú typ merania na každom kanále a hodnota všetkých naväzujúcich nastavení. Pred tým ako začne akvizícia dát, je nutné, aby sa vytvoril súbor ako pre uloženie parametrov daného merania do hlavičky TDMS súboru, tak aj na logovanie dát akvizičného mechanizmu. Na akvizíciu a logovanie dát je nutné vytvoriť virtuálne kanály v prostredí LabVIEW, ktorým budú prislúchať relevantné nastavenia a parametre. Tento proces bude podrobnejšie objasnený v ďalších troch odstavcoch a náhľad na vizuálnu interpretáciu možno vidieť na Obr.č.2.2.

#### Uloženie nastavení do hlavičky TDMS

V tomto bode je potreba zadať trasu, kde sa uloží TDMS súbor. Riešenie môže byť otvorenie okna pre určenie destinácie novo vytvoreného súboru, poprípade vrámci nastavenia parametrov môže nastať dopyt po zadaní trasy k priečinku kde chceme vytvoriť logovací TDMS súbor v procese nastavovania ostatných parametrov merania.

## **Vytvorenie kanálov**

K virtuálnym kanálom sa programovo nalinkujú všetky potrebné parametre, keďže funkcia DAQmx Create Virtual Channel má špecifickú inštanciu pre každý typ merania, kde sa pri každej prispôbia jej vstupy a výstupy. Pre potreby našej aplikácie budeme potrebovať vytvoriť 4 virtuálne kanály. Tento proces prebehne sekvenčne či už za sebou, alebo v opakujúcej sa slučke s o čosi viac premysleným mechanizmom.

## **Akvizičná slučka, logovanie dát**

V tomto bode môže meranie prejsť do slučky, ktorej iterácia bude znamenať výčet jedného bodu z každého aktívneho kanálu. Tento výčet bude tkvieť z čítania dát pomocou funkcie „DAQmx Read“. Následne by bolo vhodné dáta zobrazit' v grafe v UI a zapísať pomocou funkcie „TDMS Write“ do TDMS súboru, ktorého hlavička by obsahovala hodnoty použitých parametrov.

## 3 Rozbor vhodnej šablóny

Pre realizáciu programu sa javí byť vhodná šablóna pre nepretržité meranie a záznam dát (Continuous Measurement and Logging template). V tejto kapitole prejdeme základnými črtami tejto šablóny a vysvetlíme na akom princípe fungujú ňou použité mechanizmy. V závere by sme mali byť schopní zhodnotiť výhody, prípadne úskalia využitia tohto kódu, ako aj mať prehľad v tom, ako by sme doň jednotlivé funkčné časti spomínané v predchádzajúcich častiach dokázali naimplementovať.

### 3.1 Mechanizmus notifikátorov (Notifiers)

Sekcia vychádza zo zdroja [42]. V prípade Obr.č.3.1 sa vytvorí notifikátor - nová referencia, náväznosťou na ktorú sa bude pristupovať k dátam prenášaným pomocou notifikátorov. Jediný zapojený vstup je „element data type“, ktorý určí aký typ dát sa bude prenášať cez referenciu notifikátorov. V tomto prípade to bude 1D pole (Array) typu Waveform(DBL<sup>1</sup>).



Obr. 3.1: funkcia „Obtain Notifier“

Notifikátorová komunikácia v rámci našej aplikácie bude prenášať zbierané data s účelom ich zobrazovania na čelnom paneli (Front Panel). Notifikátorový mechanizmus na prenos dát je potencionálne stratový, preto sa pre logovanie zbieraných dát používajú rady (Queues) a notifikátory majú iba funkciu viacmenej približnej vizualizácie dát.

Slučka pre záznam dát má v sebe Case štruktúru (Case structure) s názvom „Message Cases“. V stave s názvom „Acquire“ sa nachádza ďalšia Case štruktúra, kde vojdeme do stavu s názvom „Acquiring“. Ten volá podprogram „Acquire.vi“ a v ňom sa referencia preniesie do funkcie s názvom „Send Notification“ vid. Obr.č.3.2. Táto funkcia je odosielateľom reálnych dát prenášaných notifikátorovým mechanizmom. Dáta v podobe správy sú napojené na vstupe, v našom prípade pod typom 1D pole typu Waveform(DBL) - súvisí s tým pod akým typom sme vytvorili referenciu funkciou Obtain Notifier Obr.č.3.1.

Slučka pre zobrazovanie dát (Data Display Loop) - referencia z funkcie Obtain Notifier ide aj do funkcie Wait on Notification Obr.č.3.3, ktorá sa nachádza v slučke

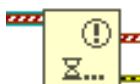
---

<sup>1</sup>Double -> Double precision floating-point format



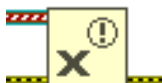
Obr. 3.2: funkcia „funkcia Send Notification“

pre zobrazovanie dát. Táto funkcia má pripojený jediný vstup a to - „notifier“, čo je referencia na notifikátor. Jeden z dvoch výstupov funkcie Wait on Notification, ktorý je použitý v našej aplikácii je „notification“, čo je v základe správa (message) ktorá sa posiela po mechanizme notifikátorov, v našom prípade v podobe 1D pola typu Waveform(DBL). Tieto dáta putujú do grafu (Waveform Chart) v slučke pre zobrazovanie dát. Wait on Notification funkcia vie na výstupe referenciu aj predať ďalej. Druhý výstup zapojený na tejto funkcii je „error out“, ktorý je zapojený na podmienkový terminál (Conditional Terminal), ktorý v prípade výskytu chyby ukončí beh slučky pre zobrazovanie dát, lebo vo svojom clusteri predá informáciu typu Boolean o hodnote True, čo indikuje chybový stav.



Obr. 3.3: funkcia Wait on Notification

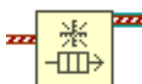
Funkcia pre vypustenie notifikátora (Release Notifier) Obr.č.3.4 odstráni referenciu pre notifikátor - v skutočnosti sa dá použiť funkcia pre získanie/vytvorenie notifikátora (Obtain Notifier) viac krát, pre získanie tej istej referencie v prípade, že do vstupu funkcie Obtain Notifier zadáme meno už existujúcej/prebiehajúcej notifikátorovej komunikácie (už existuje referencia). V tom prípade treba zavolať funkciu Release Notifier toľko krát, koľko krát sme vytvorili kópiu hoci tej istej referencie takže počet volaní funkcií Obtain Notifier by za daných podmienok bol rovný počtu volaní Release Notifier funkcie.



Obr. 3.4: funkcia Release Notifier

## 3.2 Mechanizmus rady (Queue)

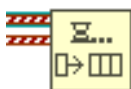
Táto sekcia vychádza zo zdroja [43]. Funkcia Obtain Queue podobne ako Obtain Notifier Obr.č.3.5 vytvorí nový rad a vo výstupe s názvom „queue out“ odošle referenciu na tento novo-vzniknutý rad (Queue). Výhody používania rád tkvejú v bezstratovom prenose dát. Existuje síce aj potencionálne stratová forma pod funkciou „Lossy Enqueue Element“ a prenášaním dát touto formou sa síce prenášajú dáta, ale nevytvárajú sa kópie týchto dát, čo je z hľadiska využívania a šetrenia pamäte a rýchlosti pochopiteľne výhodný fakt.



Obr. 3.5: Obtain Queue

Funkcia v našom prípade vytvorí nový rad, ktorý bude médiom na prenášanie a uchovávanie dát 1D poľa typu Waveform. Na výstupe odošle referenciu na novo vytvorený rad. S touto referenciou budú v našej aplikácii pracovať funkcie Enqueue Element, Dequeue Element, Flush Queue a Release Queue.

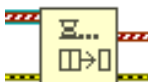
V „Acquisition Message Loop.vi“ -> Message Cases -> „Acquire“ -> „Acquiring“ -> Acquire.vi, sa nachádza funkcia Enqueue Element Obr.č.3.6. Táto funkcia pri naplnení rady v našom prípade čaká, preto nedôjde k strate dát. Čakanie funkcie na uvoľnenie rady je zaistené v inpute „timeout in ms“, ktorá je defaultne nastavená na -1, ako aj v našom prípade a znamená to, že k timeoutu nikdy nedôjde. Funkcia pridáva prvok na koniec rady, pod referenciou, ktorá je funkcii predaná.



Obr. 3.6: Enqueue Element

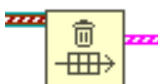
„Logging Message Loop.vi“ - Message Cases : „Log“ -> „Logging“ -> „Log Data.vi“ -> Dequeue Element funkcia Obr.č.3.7. Timeout tejto funkcie naopak čaká na moment, kedy je rada prázdna a funkcia nemá aký prvok z rady uvoľniť. Na výstupe odosiela prvok, ktorý z rady uvoľňuje (vychádza zo slova dequeue). My tento prvok následne zapíšeme do TDMS súboru. Funkcia vie taktiež odoslať referenciu na danú radu v ďalšom z výstupov, čo v našom konkrétnom prípade ale nevyužívame.

„Logging Message Loop.vi“ - Message Cases : „Stop“ -> „Logging“ -> „Flush Data to Log File.vi“ alebo Message Cases : „Exit“ -> „Logging“ -> „Flush Data



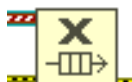
Obr. 3.7: Dequeue Element

to Log File.vi“ -> Flush Queue funkcia Obr.č.3.8, zoberie všetky elementy, ktoré sa akurát nachádzajú v rade s danou referenciou a odošle ich vo výstupe s názvom „remaining elements“ vo forme poľa (Array). Rada teda zostáva prázdna a akcia je ekvivalentná okrem predania prvkov ďalej aj odstráneniu všetkých elementov z danej rady. Funkcia vie taktiež odoslať referenciu ďalej vo výstupe „queue out“.



Obr. 3.8: Flush Queue

Release Queue funkcia Obr.č.3.9 - odstráni referenciu na danú radu. Aj táto funkcia má možnosť rovnako ako funkcia Flush Queue, ktorou je získanie všetkých prvkov v rade pred tým ako sa daná rada odstráni, v tomto prípade aj referencia. Je tu možnosť s názvom „force destroy?“ na vstupe funkcie, ktorou vieme naraz odstrániť všetky referencie pre danú radu a nie je potreba volať funkciu Release Queue toľko krát, koľko referencií bolo vytvorených. Podobne je tomu aj v mechanizme notifikátorov.



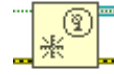
Obr. 3.9: Release Queue

### 3.3 User Event - udalosť

V tejto sekcii čerpáme informácie zo zdroja [44]. Podstatnou časťou podprogramu (subVI) „Create User Event - Stop.vi“ vid. Obr.č.3.10a je funkcia „Create User Event“ (Obr.č.3.10b), ktorá vytvorí začiatok mechanizmu. Výstupom ako funkcie, tak tohto VI, je referencia užívateľskej udalosti (User Event). Tento mechanizmus aplikácie bude slúžiť na zastavenie slučky „Event Handling Loop“ pri stlačení tlačidla „Exit“, ktorý sa nachádza na hlavnom čelnom paneli.



(a) Create User Event - Stop.vi



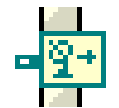
(b) Create User Event

Obr. 3.10: Vytvorenie udalosti (Event)

V predchádzajúcom odstavci som načal funkcionality mechanizmu s názvom Dynamic registration. Každá Event štruktúra využívajúca tento mechanizmus má jeden samostatný zoznam pre registráciu udalostí (Reg Events) vid. Obr.č.3.11a, ktorý môže obsahovať ľubovoľný počet udalostí. Hlavným výstupom je „event reg ref-num out“, ktorý môže ísť do ďalšieho zoznamu pre registráciu udalostí, do prvku „Event Dynamic Registration“ (Obr.č.3.11b) na okraji Event slučky alebo do funkcie „Unregister For Events“ Obr.č.3.12. V podstate sa vytvorí referencia, pod ktorou je možné mať viacero udalostí (Events) a spravovať ich v danej Event štruktúre.



(a) Register for Events



(b) Event Dynamic Registration

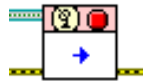
Obr. 3.11: Mechanizmus „Dynamic registration“



Obr. 3.12: Unregister For Events

V „UI Message Loop“ -> „Message Cases“ nájdeme stav s názvom „Exit“, do ktorého sa dostaneme po stlačení tlačidla Exit pomocou mechanizmu správ „Message Queue“, ktorými riadime stavy v jednotlivých slučkách. Tu sa vrámci užívateľskej udalosti vykoná „Fire User Event - stop.vi“ Obr.č.3.13a, ktorý má v zásade rovnakú štruktúru ako „Create User Event - Stop.vi“, ale pracuje s funkciou s názvom „Generate User Event“ vid. Obr.č.3.13b. Táto funkcia dostáva referenciu „User Event - Stop“, ktorej názov sme vytvorili na výstupe „Create User Event - Stop.vi“. Následne vyvolá akciu v slučke pre spracovanie udalostí (Event Handling Loop), kde sa nachádza stav s názvom „User Event“, ktorý sa generovaním udalosti automaticky spustí a v tomto prípade ukončí slučku pre spracovanie udalostí. „Fire User Event - stop.vi“ sa vykoná až po tom, čo ostatné slučky dostali správu

o reťazcovej hodnote „Exit“, čím sa ukončuje beh aplikácie. Odhliadnuc od spomenutého sa vpíše reťazec o string hodnote „Exit“ do lokálnej premennej „Status“, ktorá predstavuje reťazcový indikátor na čelnom paneli pre informovanie užívateľa o stave v ktorom sa aplikácia nachádza.



(a) Fire User Event - stop.vi



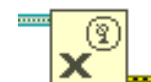
(b) Generate User Event

Obr. 3.13: Odoslanie udalosti

Po ukončení slučky putuje referencia „event reg refnum out“ do funkcie „Unregister For Events“ viď. Obr.č.3.12, kde sa zmaže. Zo subVI „Fire User Event - stop.vi“ pokračuje reťazec „error out“ cez zopár ďalších mechanizmov, až kým nedôjde do subVI „Destroy User Event - stop.vi“ Obr.č.3.14a, ktorý v sebe zahŕňa funkciu „Destroy User Event“ Obr.č.3.14b, ktorá odstráni referenciu na užívateľskú udalosť „User Event“ vytvorenú funkciou „Create User Event“. V tomto bode sa zároveň aplikácia dostáva na praktický koniec dátovej akvizície, čo sa po stlačení tlačidla „Exit“ aj očakáva. Reťazec „error out“ následne putuje do ďalšej časti programu pre korektné ukončenie chodu aplikácie.



(a) Destroy User Event - stop.vi



(b) Destroy User Event

Obr. 3.14: Odstránenie udalosti

## 3.4 Message queues

### Create All Message queues

Nesledovná sekcia vychádza zo zdroja [45]. SubVI „Obtain Message Queue.vi“ viď. Obr.č.3.15a vytvorí jednu radu typu „Message cluster“. Tento cluster je zadaný v typovej definícii (Type definition) a tvorí ho správa (Message) typu string a dáta (Message Data) typu variant Obr.č.3.15b. Funkcia Obtain Queue získava Message cluster ako „element data type (input)“. Na výstupe je referencia na radu ktorej každý jeden prvok predstavuje inštanciu čo do typu obsahu rovnakú ako Message cluster.





(a) Obtain Message Queue

**Message Cluster**



(b) Message cluster - open

Obr. 3.15: Vytvorenie riadiacej rady (Queue) pre prenos správ

SubVI „Enqueue Message (single).vi“ Obr.č.3.16a zaradí do existujúcej rady nový prvok. Na vstupe VI potrebuje referenciu na danú radu, do ktorej má pridať prvok. Na samostatných inputoch sa nachádza Message a Message Data. VI poskytuje možnosť v podobe zaradenia prioritnej správy, kedy sa použije funkcia „Enqueue Element At Opposite End“ Obr.č.3.16b a správa sa dostane na začiatok rady. V opačnom prípade sa implicitne používa funkcia „Enqueue Element“ pre klasické zaradenie prvku na jej koniec.



(a) Enqueue Message (single).vi



(b) Enqueue Element At Opposite End

Obr. 3.16: Pridanie prvku do rady

Konkrétne v prípade subVI „Create All Message Queues.vi“ Obr.č.3.17 sa nejedná o prioritnú správu a všetky tri novo vytvorené rady získavajú správu „Initialize“, ktorá neskôr nastaví počiatočné stavy v Case štruktúrach odpovedajúcich slučiek. Do súhrnu stavov v jednotlivých štruktúrach a slučkách možno nahliadnuť na Obr.č.3.18 a Obr.č.3.19.



Obr. 3.17: Create All Message Queues.vi

Predmetom tohto VI je vytvorenie rád pre správy (Message Queues), konkrétne troch: UI, Acquisition a Logging. Vo všetkých mechanizmoch sa na vytváranie týchto Queues stretávame s typovou definíciou, čo znamená, že pridanie ďalšej rady je možné a do maximálnej miery uľahčené. V aplikácii sa prakticky nachádza 5 slučiek:

Acquisition Message Loop	Logging Message Loop
"Initialize" "Start" "Acquire" "Stop" "Update Settings" "Exit" "---Copy This Frame---" Default	"Initialize" "Start" "Stop" "Upadte Settings" "Exit" "---Copy This Frame---" Default

Obr. 3.18: Stavby jednotlivých štruktúr v akvizičnej a logovacej slučke

- 1.) Event Handling Loop
- 2.) UI Message Loop
- 3.) Acquisition Message Loop
- 4.) Logging Message Loop
- 5.) Data Display Loop

- 1.) Slučka ovládaná udalosťami, napríklad tlačidlá, aktivita používateľa (uzavretie okna)...
- 2.) 3.) 4.) Sú všetko slučky s Case štruktúrami, ktoré využívajú práve mechanizmus rád so správami (Message Queues) pre určenie stavu v ktorom sa Case štruktúra bude nachádzať.
- 5.) Slučka bez pridanej štruktúry - len na vyobrazenie dát v grafe (Waveform Chart) pomocou mechanizmu notifikácií.

V sub VI „Create All Message Queues.vi“ sa teda vytvoria a inicializujú 3 rady (Message Queues), ktoré budú slúžiť na ovládanie stavov spomínaných troch slučiek.

Zo subIV „Create All message Queues.vi“ vychádza cluster pod typovou definíciou, ktorý obsahuje 3 referencie reflektujúce rady správ, ktoré rozbalí práve „Unbundle By Name“ funkcia viď. Obr.č.3.20. V tomto bode je jej výskyt nutný len pre potreby slučky užívateľských udalostí (Event Handling Loop), kde putuje len referencia „UI Message Queue“. UI Message Loop, Acquisition Message Loop a Logging Message Loop všetky získavajú prístup ku všetkým trom referenciám. Treba ale podotknúť, že Acquisition Message Loop použije takmer hneď na vstupe „Unbundle By Name“ funkciu a využívať bude iba „UI Message Queue“ a „Acquisition Message Queue“

Event Handling Loop	UI Message Loop
[0] "Start": Value Change [1] "Settings": Value Change [2] "Stop": Value Cheange [3] Panel Close? [4] Application Instance Close [5] "Exit": Value Change [6] <User Event - Stop>: User Event	"Initialize" "Init UI Refs" "Update Status" "Start" "Stop" "Acquisition Stopped" "Logging Stopped" "Broadcast settings" "Launch Settings Dialog" "Error" "Confirm Quit" "Exit" "---Copy This Frame---" Default

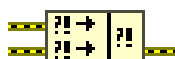
Obr. 3.19: Stavby Event štruktúry a Message Cases užívateľského rozhrania

referenciu a Logging Message Loop to má nápodobne, pričom využije UI a Logging referenciu.



Obr. 3.20: funkcia „Unbundle By Name“ rozvetvujúca referencie rád (Queues)

Funkcia „Merge Errors“ Obr.č.3.21 spojí Error cluster z „Reg Events“ funkcie a z „Create All Message Queues.vi“. Funkcia vo výstupe vráti prvú chybu ktorú nájde, pričom ju detekuje zaradom od prvého vstupu z hora. Ak chybu nenájde, vracia prvé upozornenie (warning).



Obr. 3.21: funkcia na spojenie chybových reťazcov „Merge Errors“

Error cluster z „Create All Message Queues.vi“ sa rozvetvuje do akvizičnej slučky

ako aj do logovacej slučky, výsledná chyba, upozornenie alebo len prázdny Error reťazec z tejto funkcie putuje do UI slučky. V každej slučke pôjde cez špecifické subVI slúžiace na spracovanie špecifických chybových stavov (Error Handler), pričom každá slučka má svoj vlastný. Error cluster z logovacej a akvizičnej slučky ide rovno do Merge Errors funkcie, z UI slučky prejde zrušením UI rady (Queue), rady pre logovanie a notifikátorového mechanizmu pre zobrazenie dát. Zo slučky pre spracovanie užívateľských udalostí prejde chybový reťazec (Error cluster) funkciou „Unregister For Events“ a následne sa reťazce spoja v jednej spoločnej Merge Errors funkcii. Na záver sa už iba zmaže referencia pre užívateľskú udalosť (User Event) a LabVIEW aplikácia korektne zastaví svoj chod cez podmienenú funkciu „Quit LabVIEW“.

## 4 Výsledky študentskej práce

### 4.1 Rozbor riešenia

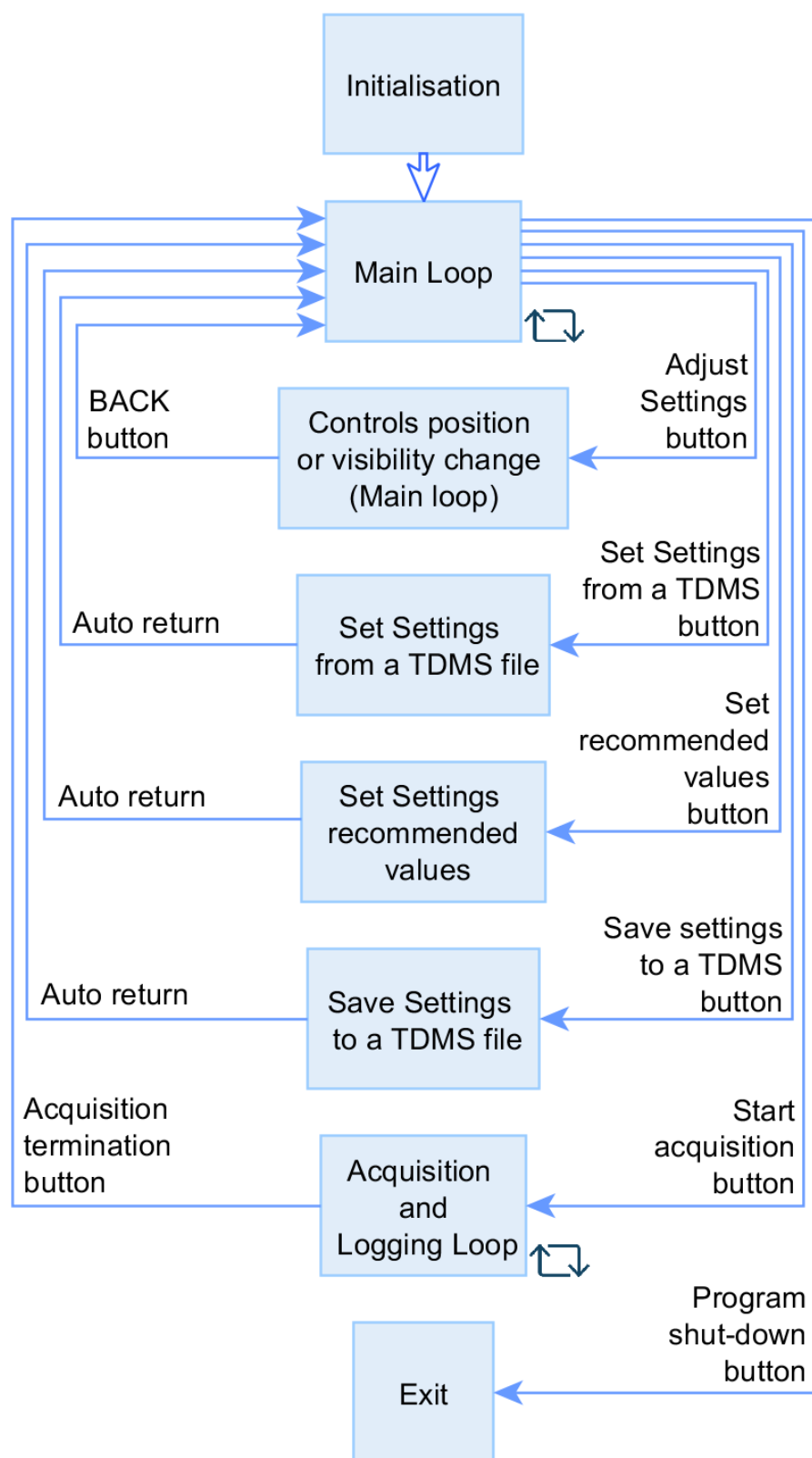
Pre prehľad a orientáciu v spomenutých a následovných kapitolách popíšem ako prebiehal celý postup práce. Na začiatku bola snaha o teoretický postup, ktorý bol vypracovaný ako prvý. Následne sme zvažovali využitie šablóny „Continuous Measurement and Logging“. Prešiel som si jej funkčnosť, ktorú som si priebežne dokumentoval. Po mnohých pokusoch písať kód rovno do šablóny som nebol úspešný a rozhodol som sa aplikáciu vytvoriť úplne celú sám na základe zadania bez akejkoľvek predlohy, s tým, že to bude neskôr ľahšie implementovateľné do spomínanej šablóny. Po vyhotovení aplikácie na to už reálne nebol dôvod, implementácia do šablóny z hľadiska očakávateľných výsledkov čo do funkčnosti z pohľadu vyhotovenia zadania nemala veľké opodstatnenie. Táto šablóna samozrejme prináša svoje výhody ako napríklad modulárnosť, rýchlosť a fakt, že ľudia pohybujúci sa v kruhoch programovania s LabVIEW alebo programovania NI produktov sa v nej ľahšie zorientujú. Práca teda pokračovala implementovaním kódu do šablóny a zhotovením druhej verzie aplikácie. Po zhotovení oboch aplikácií som otestoval ich funkčnosť na testovacím meraní.

### 4.2 Mnemosyne

V priebehu textu taktiež označovaná ako aplikácia so sekvenčnou architektúrou. Dôvod pre toto označenie spadá pod fakt, že akvizíciu dát nasleduje vykreslenie a zápis dát, všetko pod záštitou jednej while slučky. V druhej inštancii, aplikácii Mnemosyn, prebieha vykresľovanie dát v grafe (Waveform chart) a zápis dát asynchrónne, cez dva rozdielne mechanizmy - notifikátory a dátovú radu (Data Queue). Zároveň je riadiaca štruktúra Mnemosyne viacmenej sekvenčná, zatiaľ čo v prípade Mnemosyn sa pohybujeme v stavoch jednotlivých štruktúr pomocou „Message Queues“.

#### 4.2.1 Rozbor kódu

Na Obr.č.4.1 môžeme vidieť jednoduché blokové schéma aplikácie Mnemosyne. Každý jeden blok okrem „Controls position or visibility change“, ktorý je súčasťou stavu s názvom Main Loop, je stav z Case štruktúry enkapsulovanej v hlavnej while slučke tohto programu.



Obr. 4.1: Blokový diagram aplikácie Mnemosyne

## Inicializácia - Initialisation

O tom, že každé spustenie programu iniciuje stav s názvom „Initialisation“ rozhoduje jedna z dvoch častí kódu mimo hlavnej while slučky - o druhej si povieme na záver rozboru kódu v sekcii „EXIT“. Uistíme sa, že do typovej definície výčtu, uoženého ako ActualState.ctl pred spustením hlavnej slučky vpíšeme hodnotu Initialisation pomocou „Property Value (Signaling)“, pričom ju pre zaistenie počiatočného spustenia reťazcom Error napojíme na while slučku ako jej input, na ktorý čaká kým sa vykoná.

Reťazec Error ako prvé prechádza cez „Pane color“, ktorá nastaví žiadanú farbu čelného panelu. Property s názvom Origin, nastaví počiatok súradnicového systému referencovaného Panu (Pane) na žiadaný bod. Property: „Front Panel Window - Panel Bounds“ nastaví vertikálnu a horizontálnu dĺžku pre čelný panel - uspôbil som ju na základe potreby užívateľského prostredia aplikácie. Následne sa budú vrámcí ako „References Enable.vi“, tak aj „References Visible.vi“ nastavovať Property Visible a Disabled potrebných prvkov na čelnom paneli. Nastavíme ich implicitnú hodnotu, ktorú by sme vždy očakávali ak zapneme program. Pomáha to najmä v prípade, že v predchádzajúcom behu programu sa program neukončil korektne a isté hodnoty pre Property Visible alebo Disabled by neboli v nami očakávanom stave. Tieto VI sú urobené spôsobom, že si pýtajú jednotlivé referencie typovo a vo forme poľa a na jedno spustenie VI sa všetky tieto referencie nastaví iba na jednu hodnotu danej Property. V prípade Property Visible by to mohlo byť napríklad iba True alebo iba False.

Controls settings inicializuje prvé štyri prvky v našom hlavnom poli clusterov, kde sú uchované všetky nastavenia pre ľubovoľné meranie. Čo sa týka stavby samotného clusteru a tohto poľa, implementovalo sa to na obraz plánu popísaného v sekcii 2.2.1. Následne sa na 0,2 sekundy rozsvieti kontrolná LEDka. Deje sa to prechodom cez akýkoľvek stav v tejto aplikácii, ktorý neobsahuje samostatnú slučku. Implementoval som to ako jednoduchú vizuálnu kontrolu toho, že program prešiel cez daný stav. Asi najhodnotnejšie využitie má táto „LED funkcia“ pri využití tlačidla „Set recommended values“, kedy užívateľ stlačí dané tlačidlo, ale v skutočnosti ani nevie, či bol daný klik platný a aplikácia prešla do žiadaného stavu. Krátke rozsvietenie tejto LED poskytne jednoduchý kontrolný mechanizmus. Na záver nariadime aplikácii aby prešla do stavu Main Loop, reťazec Error prejde cez „Simple Error Handler“ pre zobrazenie chybového stavu a aplikácia končí svoju prvú iteráciu hlavnej while slučky.

## Main Loop

Main Loop je jedna z dvoch smyčiek viď Obr.č.4.1, v ktorej sa program nachádza, počas ktorých čaká na pokyn užívateľa - stlačenie tlačidla. V tomto stave má užívateľ k dispozícii pomocou vhodného tlačidla vojsť do akéhokoľvek stavu aplikácie ako môžeme vidieť na blokovom schémate (okrem stavu Initialisation, ktorý funkčne nemá dôvod byť v tejto možnosti).

Kód tohto stavu je enkapsulovaný vo Flat Sequence a ako môžeme vidieť, hlavná slučka iteruje pokým ju nezastaví chyba alebo interakcia používateľa v podobe stisku tlačidla. Tieto iterácie zamestnávajú procesor a neustále kontrolujú a prechádzajú cez kód v jej vnútri - polling. Ideálnym riešením by bolo zaviesť miesto celej tejto slučky Event štruktúru. Event štruktúra využíva prerušenia (ang. interrupt) - podobné funkcii myšky vrámci nášho OS pre PC, pri žiadnej interakcii nenastáva vyťažovanie procesoru a to nastane až v danej interakcii - napríklad pohybe myšky alebo v našej aplikácii stlačenie tlačidla, ktoré je zaregistrované ako udalosť (Event) v našej Event štruktúre. Problém nastáva v stave „Acquisition and Logging Loop“, kde nastáva zásah do užívateľského prostredia a tak isto interakcia užívateľa, pričom táto Event štruktúra narušovala očakávaný priebeh spomínanej slučky. Na úkor pollingu procesora sme získali istú funkcionálnu počas akvizície, ktorá nám sprístupní náhľady na nastavenia, čo sa môže v praxi využívaní aplikácie hodiť. V aplikácii Mnemasyn táto funkcionálna nebude a čakanie na interakciu užívateľa je vyriešené pomocou Event štruktúry.

Flat sequence na druhú stranu na prvý pohľad nemá opodstatnenie, keďže sú funkcie a VI dôkladne prepojené Error reťazcom pre sekvenčný priebeh. Odozva užívateľského prostredia ale vykazovala pomalšiu odozvu pri jej neprítomnosti. Z tohto jednoduchého dôvodu som ju tam nechal. Flat sequence je všeobecne považovaný za zjednodušujúci nástroj, ktorý nemá opodstatnenie (je vždy nahraditeľný vhodnejším mechanizmom), až na benchmarking - kde pomocou flat sequence vieme získať funkciu, ktorú iným pôsobom nie. Tou je v tomto prípade zisťovanie dĺžky vykonávania istej časti kódu [46]. Pre našu aplikáciu nie je nutné zistiť pravú príčinu pre tento mediálny rozdiel odozvy, keďže jej prítomnosť nečiní problém. Na druhú stranu je vhodné spomenúť, že to môže súvisieť práve s touto vlastnosťou flat sequence.

V tomto odstavci si povieme čo sa deje vo while loope stavu Main Loop. Spomenutým pollingovým systémom tu pracuje istý počet riadiacich prvkov - controls, ktoré z istého hľadiska možno rozdeliť do dvoch skupín. Tie, ktoré slučku ukončia a ich následkom je prechod do ďalšieho stavu a tie, ktoré istým spôsobom menia



UI a užívateľ ostáva prítomný v slučke, pričom zmenil hodnotu, viditeľnosť alebo umiestnenie požadovaných entít. Prvá skupina je jasná, jedná sa o tlačidlá na čelnom paneli, zlučujú sa v „OR - Compound Arithmetic“ hneď vedľa Loop Condition.

Druhú skupinu tvoria:

- 1.) Meas. Group (Radio Control)
- 2.) Adjust settings (Button)
- 3.) SubVI „Visibility Disabled Main.vi“

**Meas. Group (Measurement Group):** V štádiu ukladania nastavení do TDMS súboru, ukladania dát z akvizície alebo načítavania dát z TDMS súboru sa čerpá z aktuálne nastavenej hodnoty tohto riadiaceho prvku. Vytvára to rýchlu možnosť pre zoskupenie jedného až troch meraní v jednom súbore s tým, že vôbec nie je využitie tohto prvku nutné. Výhodou použitia tohto prvku je rýchlosť a jednoduchosť využitia.

**Adjust settings:** Stlačením tohto tlačidla sa bežiaca aplikácia posunie na čelnom paneli doprava a užívateľovi sa poskytne náhľad na iné kontrolné prvky. V tejto Case štruktúre sa okrem posunu isté prvky pomocou Property Visible zakryjú, iné sa užívateľovi poskytnú, hlavnou funkciou tohto posunu je v podstate oddelenie riadiacej časti od nastavovacej časti, kde pri každej sekcii možno očakávať iba relevantné prvky.

**Visibility Disabled Main.vi:** Toto VI má na starosti výčet pre voľbu kanálu (Channel), výber typu merania na každom z kanálov (Meas. type for AI0-AI3) a pole clusterov s nastaveniami (Controls settings). Všetky tieto ovládacie prvky v riadiacej či nastavovacej časti čelného panelu sú korigované z tohto VI. Ovládajú Property Disabled a Property Visible. Jednotlivé parametre v Controls settings sú viditeľné na základe toho, na ktorý kanál aktívne prihliadame a nastavujeme ho (teda, ktorý je v aktivovaný, v stave Enabled) a na základe toho aký typ merania je na ňom aktuálne nastavený. Ku všetkým týmto prvkom pristupujeme pomocou Property Controls[ ] ktorá vracia referencie na prvky daného Panu (Pane), ktorý je zapojený na jej vstupe. Pre nás je spomenutý Pane v podstate náš jediný aktívny Pane, ktorým je zároveň celý čelný panel.

Následne do MeasTypeMainVis.vi ideme s tým, že je jasné, o aký typ merania v danom kanále pôjde. Celkový počet parametrov v Controls settings je 29, čo predstavuje cluster s označením prvkov od 0-28. Keďže vieme na akej pozícii sa nachádza aký parameter, pre každý typ merania môžeme rozhodnúť či sa zobrazí alebo nie tak,

že bude každé jedno číslo od 0-28 prechádzať Case štruktúrou, ktorá pridelí každému číslu, respektíve každej referencii hodnotu True alebo False a následne to rozhodne o tom, či bude daný parameter viditeľný - čo už konkrétne vykonáva „Visible Ref type.vi“ pre každý jeden z 29 parametrov.

### **Nastavenie parametrov zo súbora: Set Settings - from a TDMS file**

Do tohto stavu sa dostane užívateľ po stlačení tlačidla „Set Settings from a TDMS“ ako naznačuje Obr.č.4.1, otvorí sa okno pre zadanie trasy pre vytvorený súbor a slučka v tomto stave prebehne 4 krát, jeden krát pre každý zo 4 kanálov. V každej iterácii sa určí o aký typ merania na danom kanále ide a tým sa určí od parametrov s akými menami bude záujem získať hodnoty pomocou TDMS Get Properties. V procese získavania týchto parametrov bola niekedy nutná typová konverzia, ktorú som riešil väčšmi špecifickou Case štruktúrou, ktorá pre daný typ odoslala konštantu o špecifickejšej hodnote do konkrétnej premennej daného parametru.

Za slučkou sme podobným spôsobom čítali časové parametre (Timing settings) pomocou TimeSettingsFromTDMS.vi. V závere prechádza reťazec Error cez subVI ErrorHandler9219\_SSfrom.vi predstavujúce ošetrovanie pre zopár častých chýb na ktoré mohol užívateľ natrafiť. V závislosti na tom, či užívateľ rozoznal danú chybu na ktorú sa ho dialóg pýta, určí, či sa daný Error odstráni, a poskytne mu šancu danú chybu ľahko napraviť, alebo oprava danej chyby navrhnutým spôsobom nefunguje a Error sa pošle ďalej do „Simple Error Handler.vi“, kde ho identifikujeme štandardným spôsobom. Túto funkcionálnosť poskytuje dialógové okno s dvomi tlačidlami, ktoré o tomto postupe rozhodne na základe reakcie užívateľa. V prípade výskytu ďalších opakovaných chýb poskytuje toto VI priestor pre naimplementovanie ďalších výnimiek prameniach z iných výnimočných situácií ku ktorým som sa do tohto bodu zatiaľ nedostal, dvere sú ale otvorené pre ďalší vývoj a rozšírenie aplikácie.

### **Nastavenie parametrov: Set Settings - recommended values**

V tomto stave sa na každom kanále v závislosti na tom, aký je na ňom nastavený typ merania, nastaví vhodné hodnoty pre všetky používané parametre.

### **Uloženie parametrov: Save Settings - to a TDMS file**

V tomto stave sa užívateľ ocitne po stlačení tlačidla „Save settings to a TDMS“. Užívateľ zadá cieľovú trasu pre nový súbor, poprípade zvolí už existujúci, kde sa dané nastavenia uložia pod danú meraciu skupinu „Measurement Group“, ktorú má aktívne označenú v Radio button s názvom „Choose a group name in which to operate“. Pod danú meraciu skupinu sa vpišu názvy pre všetky kanály karty a

prisluchujúce typy meraní zadané užívateľom. V úvode sa tak isto vytvoria prvky reprezentujúce každý jeden cluster súvisiaci s jedným zo 4 kanálov meracej karty. Do hlavičky súboru (pod žiadnou skupinou alebo kanálom TDMS súboru) sa vpíše dátum a čas.

Následne sa dostávame do slučky, ktorá prebehne 4 krát. Popíšeme si jednu iteráciu. V prvej iterácii sa priradí referencia clusteru (element) súviaceho s kanálom prvého analogového vstupu AI0 do druhej Case štruktúry, kde sa nachádza subVI SStoTDMS.vi. O stave v ktorom bude druhá Case štruktúra rozhodne vybraný typ merania na kanále AI0 v tomto prípade. Spomínané VI dostane referenciu na príslušný cluster v ktorom sa nachádzajú parametre ku ktorým chce pristupovať, Group a Channel name, ktoré zaradia parametre do vhodných sekcií hlavičky a ako najdôležitejší prvok vystupuje pole 32-bit integerov - teda pole čísel, ktoré budú slúžiť ako výčet parametrov, ktoré chceme ukladať do určeného TDMS súboru.

Slučka vo vnútri SStoTDMS.vi sa vykoná toľko krát, koľko prvkov má v sebe pole celých čísel súvisiaci s typom merania, ktoré bude vykonávané na danom kanále. Každé číslo reprezentuje istý parameter, pričom každý môže byť iného typu a bude potreba mať iný spôsob ako ho do súboru zapísať. Na to slúži druhá Case štruktúra v tejto slučke, ktorá dostane informáciu o type parametru, ktorý ide zapisovať. Vhodným spôsobom teda pristúpi k hodnote daného control/parametru a vpíše ju do TDMS súboru pomocou funkcie TDMS Set Properties.

V záverečnej časti je tomu podobne ako v už spomínaných stavoch. Zapišu sa aj hodnoty Timing Settings a v závere je tu ďalší Error handler, ktorý môže byť použitý pre špecifické chyby, ktoré sa vyskytli iba v tejto časti kódu a nie je potreba ich ošetrovať v iných stavoch.

### **Akvizičná a logovacia slučka: Acquisition and Logging Loop**

Prvá slučka tohto stavu inicializuje viditeľnosť všetkých štyroch kanálov pomocou Property „Active Plot“. Druhá slučka vytvorí pomocou subVI „DAQmx Create Channel“ daného typu kanál pre potrebné meranie. Vo vhodnom stave Case štruktúry teda dôjde aj k priradeniu všetkých potrebných parametrov k virtuálnemu kanálu. Za slučkou sa vhodným spôsobom upraví časť UI (Property: Disabled, Visible), zbytok bude pozmenený vrámci tretej slučky ku ktorej sa čoskoro dostaneme. Rovnako sa zapišu hodnoty časových parametrov (Timing settings) a čo do funkčnosti sme splnili všetko potrebné pre začatie procesu akvizície dát.

SubVI „Visibility Disabled AcqLog.vi“ je veľmi podobné už raz opisovanému VI, ktoré sa nachádza v Main Loop pod názvom „Visibility Disabled Main.vi“. Rozdiel medzi týmito VI je, že to, ktoré sa nachádza v akvizičnej slučke nedovoľuje užívateľovi meniť dané nastavenia. Všetky viditeľné kontrolné prvky sú znefunkčnené, ich funkciou je prehľad o hodnotách nastavených parametrov či už v rámci časových nastavení, settings clusterov pre jednotlivé kanály alebo typov meraní. Tieto 3 možnosti je možné jednotlivo zobrazovať počas akvizície prepínaním Radio Button pod názvom „Show:“. Podmienky, ktorým budú jednotlivé prvky podliehať sú naprogramované v Case štruktúre akvizičnej slučky riadenej práve spomínaným prvkom Radio Button. „References Visible.vi“ pracuje na rovnakom princípe ako „References Enable.vi“, obe hromadne cez pole na vstupoch jednotlivo menia podľa typu daných referencií danú Property: Visible alebo Disabled. Slučka obsahuje vlastný Error handler ako sme sa s tým mohli stretnúť v prípade stavov „Save Settings - to a TDMS file“ a „Set Settings - from a TDMS file“.

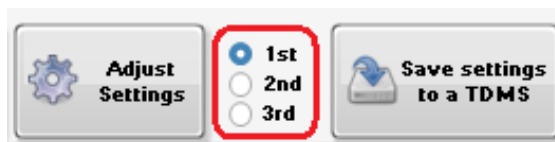
### **Ukončenie programu - EXIT**

Po stisku „Program shut-down“ sa ocitneme v poslednom stave s názvom „EXIT“. V tomto stave posielame hodnotu True do „Loop Condition“ našej hlavnej slučky v ktorej je enkapsulovaný všetok funkčný kód. Po ukončení behu slučky sa spustí posledná funkcia „Quit LabVIEW“, ktorá korektne ukončí chod bežiackej aplikácie.

### **4.2.2 Návod na použitie aplikácie**

Vzhľadom na veľký rozsah textu, ktorý by táto sekcia vygenerovala som sa rozhodol nahráť krátke demo s ukážkou funkcie aplikácie. Pre stručnosť a čo najdôkladnejšie pochopenie na čas venovaný danej činnosti som urobil demo iba pre aplikáciu Mne-masyn. Demo je zahrnuté v priloženom CD, ktorého hierarchiu a teda aj umiestnenie dema možno vidieť v kapitole 7: Obsah priloženého CD. Čo do funkčnosti boli rovnakým spôsobom testované obe aplikácie a vo výsledku neboli spozorované rozdiely. Čo som nemal ako otestovať je to, ako jednoduché je pre nového užívateľa používať túto aplikáciu. Ak by sa preukázalo, že je UI dostatočne intuitívne na okamžitú obsluhu, môžeme priložené demá ignorovať. V nasledujúcich odstavcoch popíšeme, čo k demu bolo nutné dodať a poukážeme na rozdiely, ktoré prichádzajú v aplikácii Mnemosyne.

V aplikácii sme počas natočenia dema nemali sprístupnenú voľbu parametru „Physical channel“ pre užívateľa z dôvodu testovania na simuláciách. Obe aplikácie, ktoré odovzdávame sú už o tento prvok obohatené a užívateľ vidí, do akého zariadenia (De-



Obr. 4.2: Voľba Measurement Group

vice<sup>1</sup>) a pod aký kanál sa dané meranie bude zapisovať. Predovšetkým ide o výber „Device“, ktorý je s pomocou „Recommended values“ nastavený vždy na „Dev1“, takže si to užívateľ môže vždy pre svoju potrebu zmeniť na iné meno, poprípade si nové zariadenie v NI-MAX vždy zaradiť pod meno „Dev1“ a všetko bude automaticky fungovať s predvolenými hodnotami.

V deme som spomenul vytvorenie kanálu „AI Voltage“, ak si užívateľ vyberie typ merania pod názvom „Nothing“. Program v tomto prípade nezapisuje parametre, každopádne by v simulácii generoval dáta odpovedajúce meraniu napätia. V reálnom meraní zaznamenáme šum, keďže na danom vstupe nebude prítomný žiaden signál. Dôvod pre nutnosť tohto riešenia bol vo vytváraní virtuálnych kanálov v akvizičnej sekcii, kedy by neprítomnosť vytvoreného kanálu spôsobovala nestále poradie vytvorených kanálov a tým aj ich nesprávne zaradenie vrámci zápisu do súboru. Ponúkané riešenie oplýva funkčnosťou a doposiaľ nebol spozorovaný výrazný problém s touto implementáciou.

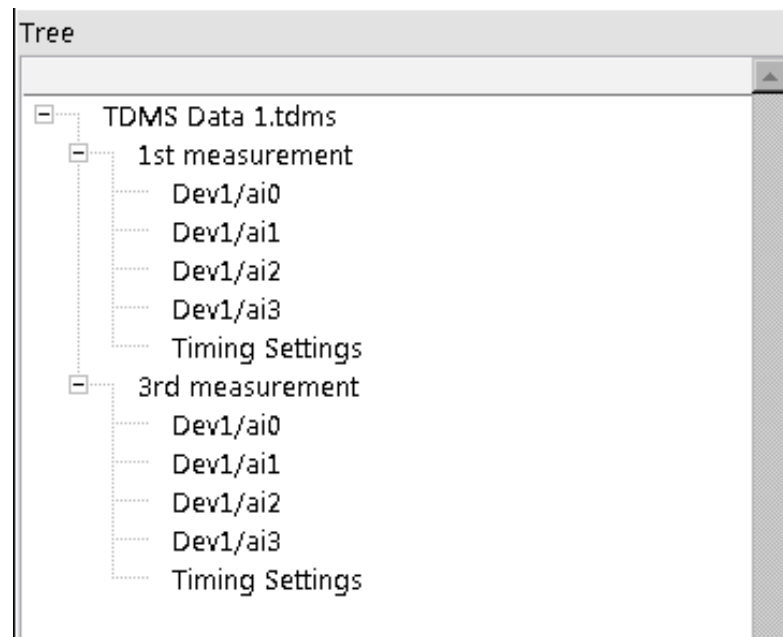
Čo sa týka rozdielov, ktoré navyše ponúka aplikácia Mnemosyne, ide predovšetkým o 3 výraznejšie aspekty:

- Zaradenie do meracích skupín (Measurement groups)
- Možnosť nahliadnutia do akýchkoľvek nastavení počas akvizície
- Možnosť samostatne vopred uložiť nastavenia - neukladajú sa automaticky spolu so začatím akvizície dát ako v aplikácii Mnemasyn

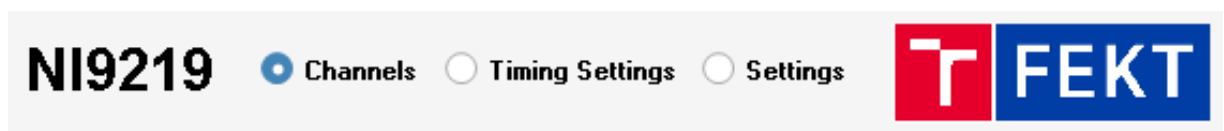
**Zaradenie do meracích skupín:** Ako vyzerá voľba Measurement group počas behu aplikácie môžeme vidieť na Obr.č.4.2. Následne sa to prejaví vo výslednom TDMS súbore napríklad ako v prípade, ktorý môžeme sledovať na Obr.č.4.3. V tomto TDMS súbore užívateľ uložil nastavenia ako do prvej, tak do tretej "Measurement group". Tento súbor má v sebe teraz potenciál dvoch samostatných súborov s rozličnými meraniami na všetkých štyroch kanáloch.

**Nahliadnutie do akvizície:** Pomocou riadiaceho prvku „Radio buttons“ z Obr.č.4.4 máme možnosť nahliadnuť do jednotlivých parametrov zvolených pre dané meranie.

<sup>1</sup>DevX; X=1,2...



Obr. 4.3: Ukážka Measurement Groups v TDMS súbore



Obr. 4.4: Ukážka riadiaceho prvku pre zobrazenie nastavení počas akvizície

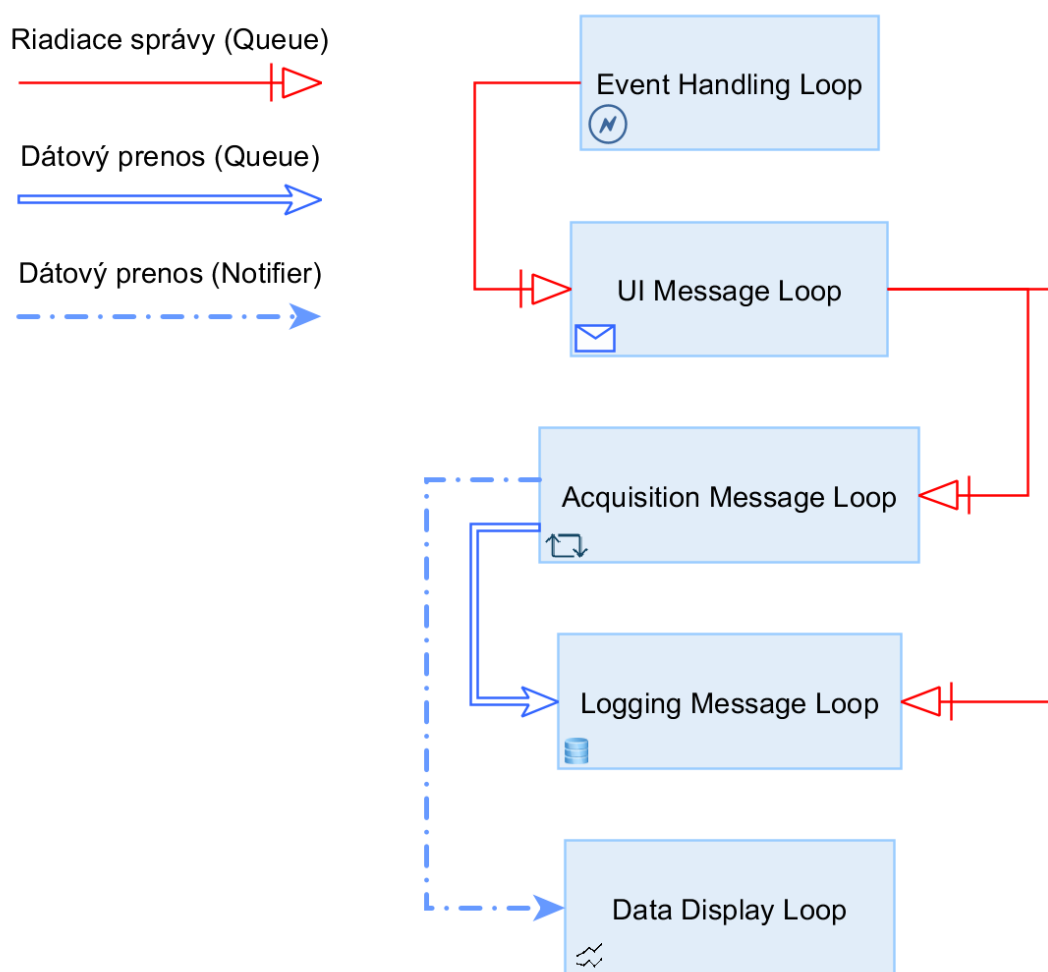
Jediný prístupný prvok na zmenu bude prepínanie medzi kanálmi, zmeny hodnôt nie sú sprístupnené.

**Samostatné uloženie parametrov merania:** Spustenie samotnej akvizície dát v aplikácii Mnemosyne súbor otvára. Pred tým ako začneme akvizíciu, chceme použiť button „Save settings to a TDMS“, určíme meraciu skupinu, v ktorej chceme zaradiť meranie a vytvoríme TDMS súbor s parametrami merania. Až potom môžeme spustiť akvizíciu a vybrať súbor do ktorého chceme uložiť namerané dáta. Na rozdiel, aplikácia Mnemosyn spustením akvizície vytvorí súbor a dátam z akvizície automaticky priradí parametre. Každá z týchto možností ponúka výhody a nevýhody, obe aplikácie to tým pádom majú implementované iným spôsobom. Objektívnym pohľadom nahliadnuc je si myslím vhodné vybrať čo by užívateľovi vyhovovalo a v tom prípade je to v pokročilejšom merítku vhodnejšie riešenie v aplikácii Mnemosyne (oddelenie uloženia parametrov a uloženia dát akvizície), hoci je to menej užívateľsky prívetivé.

## 4.3 Mnemasyn

### 4.3.1 Rozbor kódu

Čo sa týka aplikácie Mnemasyn, povenujeme sa vyslovene rozdielom medzi jednotlivými aplikáciami a prvkom, ktoré som doteraz nespomenul. Samotným mechanizmom, týkajúcich sa šablóny „Continuous Measurement and Logging“ som sa venoval v sekcii 3 a vlastnému riešeniu, ktoré som do šablóny nakoniec implementoval, som sa venoval v sekcii 4.2. V priebehu implementácie ale došlo k istým zmenám a práve týmito zmenám sa bude venovať táto sekcia.



Obr. 4.5: Blokové schéma aplikácie Mnemasyn [47]

Naprieč celým kódom bolo mojim zámerom ho nechať čo najviac pôvodným a nemeniť mnoho prvkov, skôr sa sústrediť na implementáciu funkčnosti, ktorú v tejto šablóne potrebujeme dosiahnuť. Výsledkom bolo aj zanechanie všetkých komentárov,

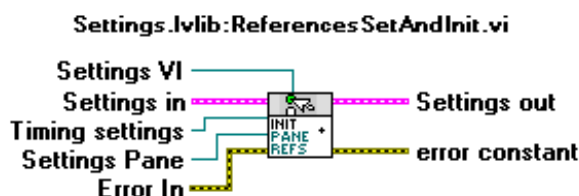
ktoré popisovali akým spôsobom môže užívateľ prispieť do šablóny pod označením napríklad #CodeRecommended. Aplikácia je tým pádom naďalej otvorená ďalšiemu rozvoju.

Blokové schéma na Obr.č.4.5 popisuje komunikáciu mechanizmov použitej šablóny. V nasledujúcich paragrafoch si ukážeme kde a ako sú naimplementované jednotlivé funkčné časti aplikácie. Istým spôsobom som potreboval naimplementovať jednotlivé stavy Case štruktúry aplikácie Mnemosyne vid. Obr.č.4.1 do spomenutej architektúry, ktorú popisuje Obr.č.4.5. Inštancie, ktoré plánujem v skratke popísať a ukázať patria do troch knižníc, objekty každej knižnice spadajú pod jednu slučku, čo značí názov každej z troch následovných sekcií. Príslušnosť konkrétneho VI do knižnice možno spoznať z jeho zaradenia v názve, ale aj v grafickej úprave hlavičky. Jednotlivé VI sa dajú ľahko vyhľadať cez projektovú štruktúru aplikácie. Projekt obsahuje 3 knižnice, po rozkliknutí môže užívateľ kliknúť na ľubovoľné VI a pravým tlačidlom rozkliknúť ponuku: Find - Callers, čím sa vie ľahko preklikať z ľubovoľne hlboko vnorenej štruktúry. Najbližších 12<sup>2</sup> popísaných VI v následovných sekciách textu poskytujú väčšinu implementovaného kódu a obsahujú takmer všetku funkčnú časť aplikácie Mnemosyn.

## UI Message Loop - slučka ktorá rozosiela riadiace správy

### Inicializácia a nastavenie referencií: Obr.č.4.6

Do zloženého pôvodne nazvaného „Config“ clusteru (prítomný na čelnom paneli) sa vložia potrebné referencie. Na prvé volanie tohto VI sa inicializuje MCS (Main Controls Settings) - hlavné pole clusterov s nastaveniami pre rôzne typy meraní a vykonajú sa náležitosti podobné stavu s názvom „Initialise“ aplikácie Mnemosyne.



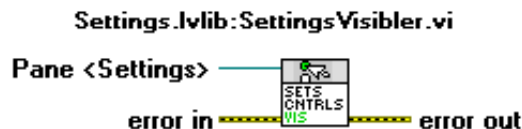
Obr. 4.6: Inicializácia a nastavenie referencií

### Nastavenie viditeľnosti jednotlivých parametrov MCS (Main Controls Settings): Obr.č.4.7

Toto vi je z funkčného hľadiska takmer totožné s VI, ktoré je popísané v sekcii 4.2.1: Visibility Disabled Main.vi:.

<sup>2</sup>po 4 z každej knižnice





Obr. 4.7: Nastavenie viditeľnosti parametrov MCS

#### Nastavenie parametrov z TDMS súboru: Obr.č.4.8

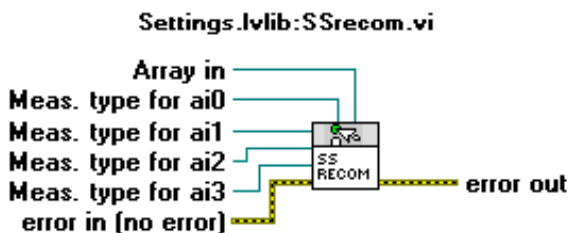
V tomto VI sa načítajú parametre do MCS a zároveň sa zo súboru vyčítajú aj Timing Settings. VI je vysvetlené z funkčného hľadiska v sekcii 4.2.1: Nastavenie parametrov zo súboru: Set Settings - from a TDMS file. Tieto VI boli prispôsobené aplikácii a majú aj rozdielne prvky, ktorých funkcionality alebo zmysel je už jednoduchšie pochopiť a nebudeme sa nimi zaoberať.



Obr. 4.8: Načítanie parametrov z TDMS súboru

#### Nastavenie odporúčaných parametrov: Obr.č.4.9

Hlavný princíp tohto VI bol vysvetlený v sekcii 4.2.1: Nastavenie parametrov: Set Settings - recommended values.

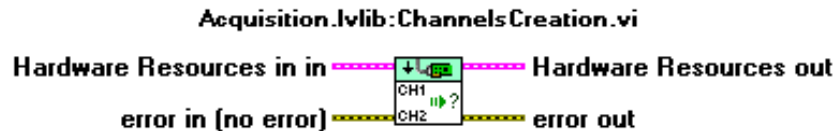


Obr. 4.9: Nastavenie odporúčaných hodnôt parametrov

### Acquisition Message Loop - Akvizičná slučka, zber dát

#### Vytvorenie virtuálnych kanálov: Obr.č.4.10

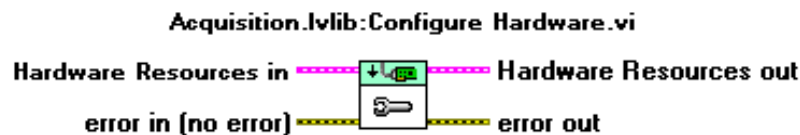
V porovnaní s vytvorením virtuálnych kanálov v sekcii 4.2.1: Akvizičná a logovacia slučka: Acquisition and Logging Loop, je zjavné, že sa v prípade tejto aplikácie spoľiehame na prácu s referenciami. Z funkčného hľadiska sa ale jednotlivé kanály vytvoria rovnako a poskytnú rovnaký finálny produkt.



Obr. 4.10: Vytvorenie virtuálnych kanálov

#### Timing settings, Start task - časové parametre, vytvorenie úlohy pre dátový zber: Obr.č.4.11

VI spadá pod sekciu, ktorú som referencoval v predchádzajúcom paragrafe. Aplikácia je modulárna a znova pristupujeme k jednotlivým akciám cez referencie. Tento spôsob vytvára prehľadnú organizáciu a ľahšie orientovanie sa v jednotlivých častiach kódu.



Obr. 4.11: Zápis Timing Settings, start task

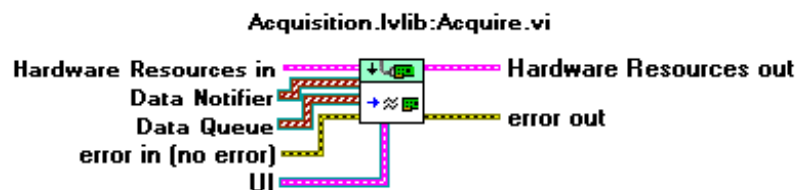
#### Akvizícia: Obr.č.4.12

Stále sa nachádzame v sekcii 4.2.1: Akvizičná a logovacia slučka: Acquisition and Logging Loop porovnávania s aplikáciou Mnemosyne. Aplikácia Mnemosyne dáta zapisovala rovno do TDMS súboru a zobrazovala ich v grafe (Wf<sup>3</sup> Chart). V tomto prípade vidíme, že sa dáta posielajú do rady (Queue) a mechanizmu notifikátorov. Mechanizmus Queue dáta odošle do logovacej slučky a notifikátorový mechanizmus odošle dáta do zobrazovacej slučky<sup>4</sup>. Queue mechanizmus je bezstratový a je teda ekvivalentný priamemu zápisu do TDMS súboru. Mechanizmus notifikátorov je možno stratový a nie je vhodný na logovanie dát, každopádne na zobrazenie v grafe je vhodný. Podrobný popis týchto dvoch mechanizmov sa nachádza v kapitole 3 v prvých dvoch podkapitolách. V tomto bode je vhodné nahliadnuť na Obr.č.4.5 a uvedomiť si tok dát naprieč slučkami - v tomto paragrafe som popisoval Dátový prenos (Queue) a Dátový prenos (Notifier) ak mám poukazovať na blokové schéma a výrazy v ňom použité. Samozrejme aj v tomto bode je pohotová Event štruktúra a na povel užívateľa sa pomocou posielania správ využíva aj mechanizmus Riadiacich správ<sup>5</sup>, taktiež v bezstratovom Queue mechanizme.

<sup>3</sup>Wf = Waveform

<sup>4</sup>Data Display Loop

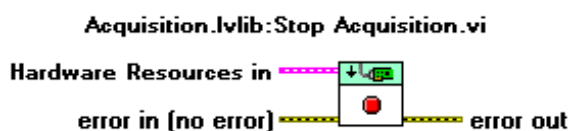
<sup>5</sup>Taktiež výraz z legendy blokového schématu Mnemosyn 4.5



Obr. 4.12: Dátová akvizícia

#### **Zastavenie dátovej akvizície:** Obr.č.4.13

VI obsahuje „DAQmx Stop Task“ a „DAQmx Clear Task“. Pre tento konkrétny prípad sa v praxi deje to, že užívateľ stlačí tlačidlo na zastavenie akvizície. Tento stav v Event štruktúre odošle správu „Stop“ do „UI Message Loop“. V stave „Stop“ v UI Message Loop sa odošlú správy v znakovom reťazci „Stop“ do logovacej a akvizičnej slučky. V akvizičnej slučke to bude predstavovať stav, kde sa nachádza práve „Stop Acquisition.vi“.



Obr. 4.13: Zastavenie akvizície

### **Logging Message Loop - logovacia slučka**

#### **Zápis parametrov do TDMS súboru:** Obr.č.4.14

VI je z funkčného hľadiska rovnaké ako stav popisovaný v sekcii 4.2.1: Uloženie parametrov: Save Settings - to a TDMS file. V logovacej slučke sme ale museli predať informáciu, že chceme najprv uložiť nastavenia a tak sme v stave „Start“ odoslali miesto „Log“ správy, správu s názvom „Save settings to TDMS“ po Logging Queue<sup>6</sup>. Tento ďalší stav som vopred vytvoril a jeho presný názov poslal po spomenutej rade pre logovaciu slučku. Následne po vykonaní SStoTDMSmother.vi a STStoTDMS.vi odošleme po Logging Queue správu „Log“ a slučka prejde do stavu, kde odoberá dáta z dátovej rady plnenej v akvizičnej slučke (stav „Log“).

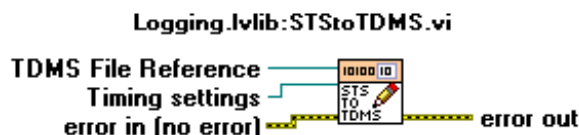
#### **Zápis timing nastavení do TDMS súboru:** Obr.č.4.15

Z hľadiska modularity bolo toto VI implementované oddelene, funkčnosťou je súčasťou stavu „Save Settings - to a TDMS file“ v sekcii 4.2.1.

<sup>6</sup>Logging Queue predstavuje mechanizmus rád (Queues) ktorou je riadená logovacia slučka; nástrojom riadenia sú znakové reťazce (String), ktoré rozhodujú o stave Case štruktúry vo vnútri logovacej slučky



Obr. 4.14: Zápis parametrov do TDMS



Obr. 4.15: Zápis timing nastavení do TDMS

#### Logovanie dát: Obr.č.4.16

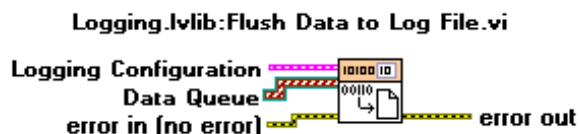
Toto VI iba odoberie prvok z rady, ktorá je plnená v akvizičnej sekcii pomocou funkcie Dequeue Element. Tento prvok potom zapíše do požadovanej „Group name“ alebo „Channel name“ TDMS súboru pomocou TDMS Write funkcie.



Obr. 4.16: Logovanie dát

#### Ukončenie logovania: Obr.č.4.17

Stavy slučky ako „Stop“, tak aj „Exit“ využijú funkciu tohto VI. Funkcia Flush Queue vymaže všetky zbytné prvky danej rady a odošle ich v podobe poľa<sup>7</sup> ako výstup, kde ich toto VI všetky zapíše do vybratého relevantného súboru a v závere daný súbor zavrie - referenciu tohto súboru už ďalej v aplikácii neposúvame.



Obr. 4.17: Ukončenie logovania

<sup>7</sup>Array

### **4.3.2 Návod na použitie aplikácie**

K aplikácii Mnemasyn je v práci priložené demo s ukážkou spustenia a vykonávania merania. V krátkosti som spomenul už v sekcii 4.2.2 čo demo prináša a aké zmeny možno spozorovať s porovnaním s aplikáciou Mnemosyne. Zároveň spomenutá sekcia ponúka všetky dodatočné informácie, ktoré bolo nutné k demu dodať.

## 5 Testovacie meranie

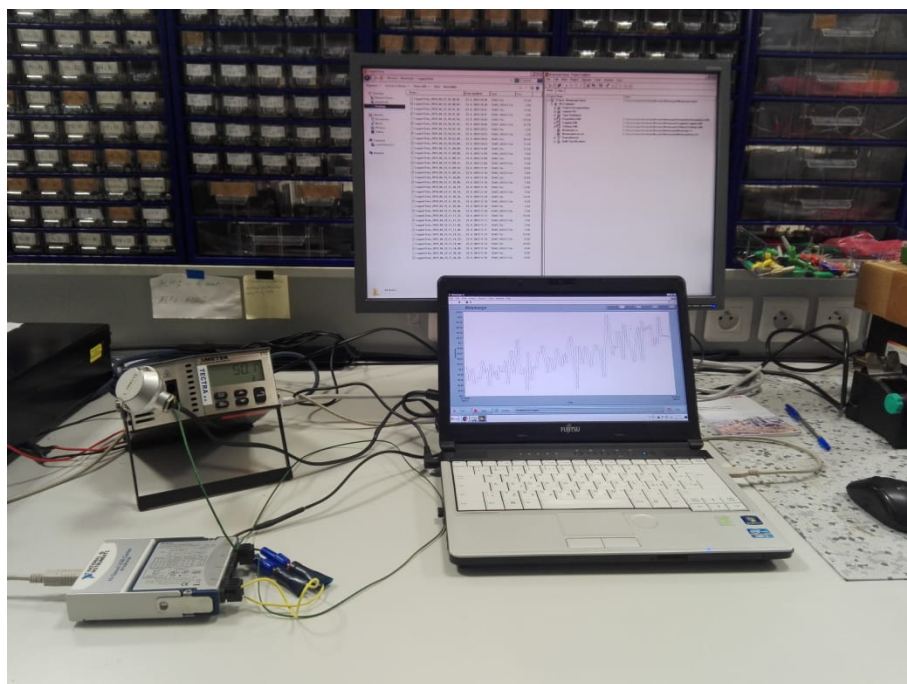
### 5.1 Návrh

Testovacie meranie dané zadáním má ako podmienku to, že sa bude merať viac veličín zároveň. Pre otestovanie som mal k dispozícii kalibračnú pec Jofra ETC-125 A, termočlánok typu K a RTD senzor. Pre otestovanie funkčnosti som zvolil teplotný skok z 10°C na 60°C. Termočlánkom budeme sledovať nameranú teplotu na AI0 a pomocou RTD senzoru budeme sledovať výsledný odpor v čase prechodu teploty referenčného meradla na vstupe AI3. Výsledné priebehy porovnáme a vyvedieme z merania záver.

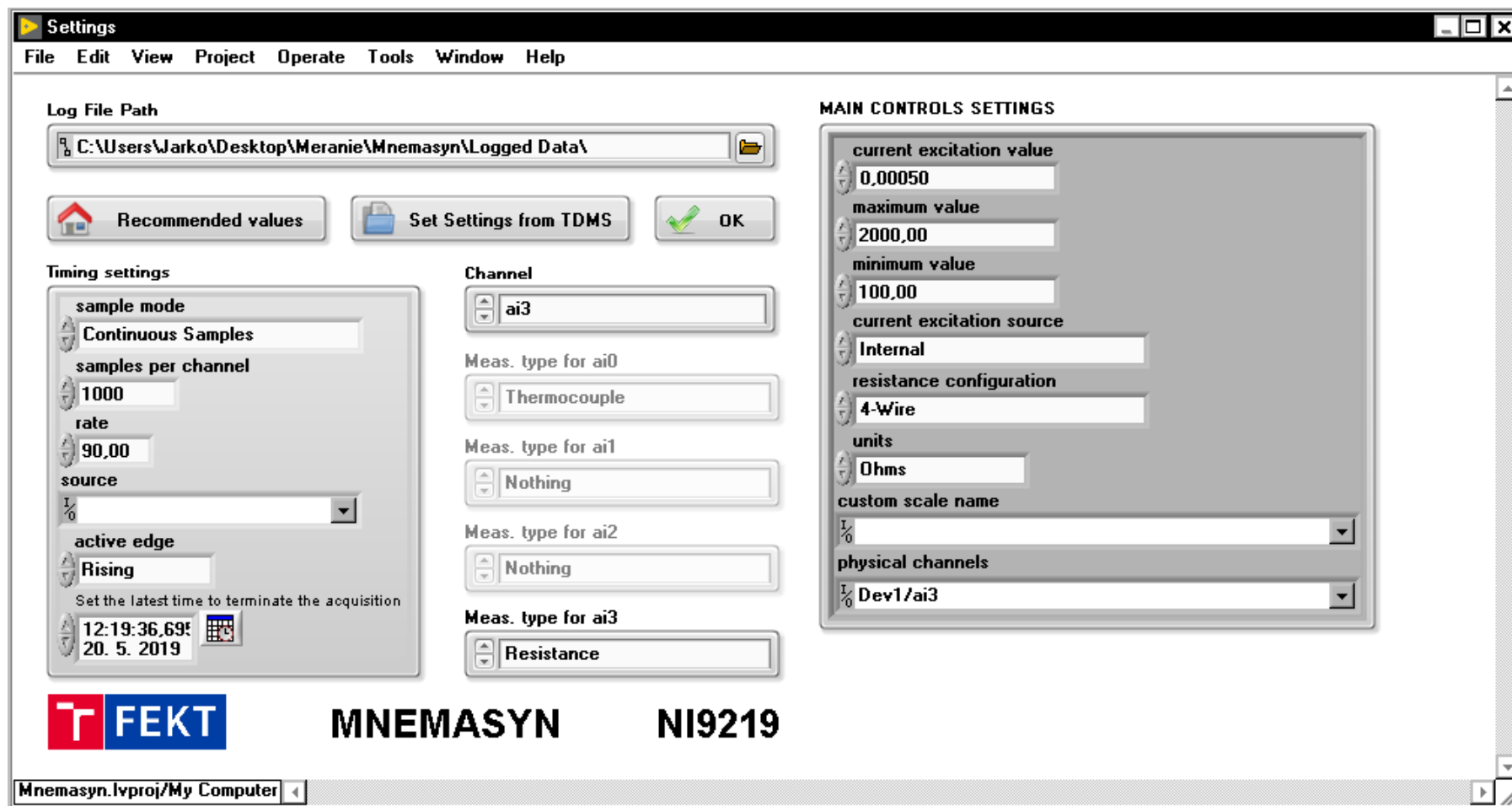
### 5.2 Riešenie

Všetky typy meraní boli pred skutočným meraním vyskúšané simuláciou. Simulácia sa nespustí, pokiaľ istý z parametrov vykazuje nelogické alebo typovo neprípustné hodnoty. Všetky typy meraní je možné spustiť simuláciou s nastavenými doporučenými hodnotami „Recommended values“ a tak isto ich možno všetky zapísať do súboru bez toho, aby aplikácia hlásila chyby. Po načítaní ľubovoľného súboru sú obe aplikácie schopné rovno spustiť meranie na základe parametrov merania, ktoré prebiehalo na tomto súbore.

Na Obr.č.5.1 sa nachádza náhľad na zostavu vrámci ktorej prebiehalo testovacie meranie. Našu kartu NI9219 máme zapojenú do NI USB-9162 nosiča, ktorý slúži na prepojenie nášho C-series modulu k PC. Komunikácia prebieha pomocou USB so špecifikáciou rozhrania 2.0 [48], takže bolo možné využiť akýkoľvek USB port dostupný na našom PC bez zmeny výsledku vo veci rýchlosti komunikácie.

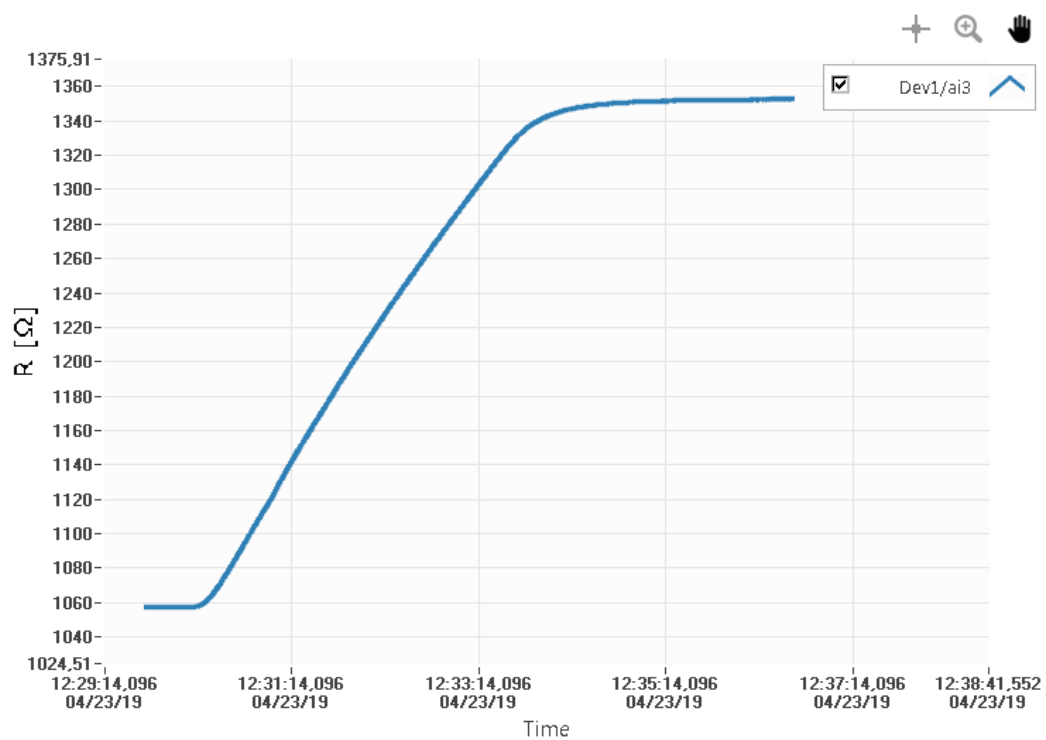


Obr. 5.1: Náhľad na zostavu testovacieho merania

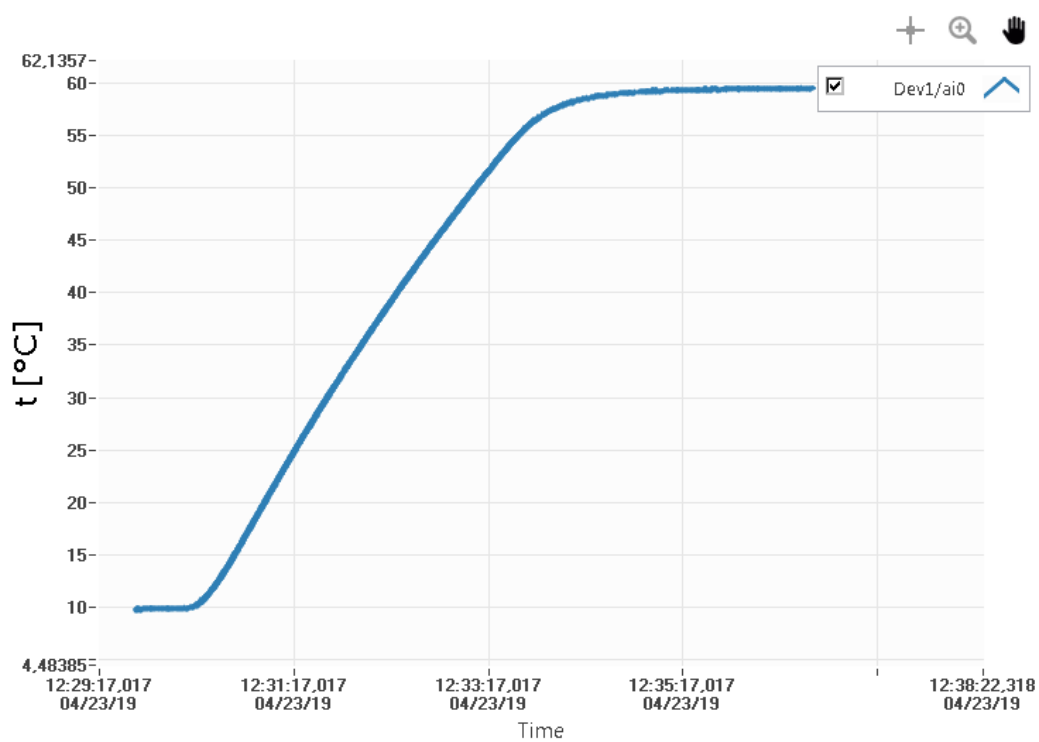


Obr. 5.2: Front panel počas merania odporu (Resistance) (Mnemasyn)

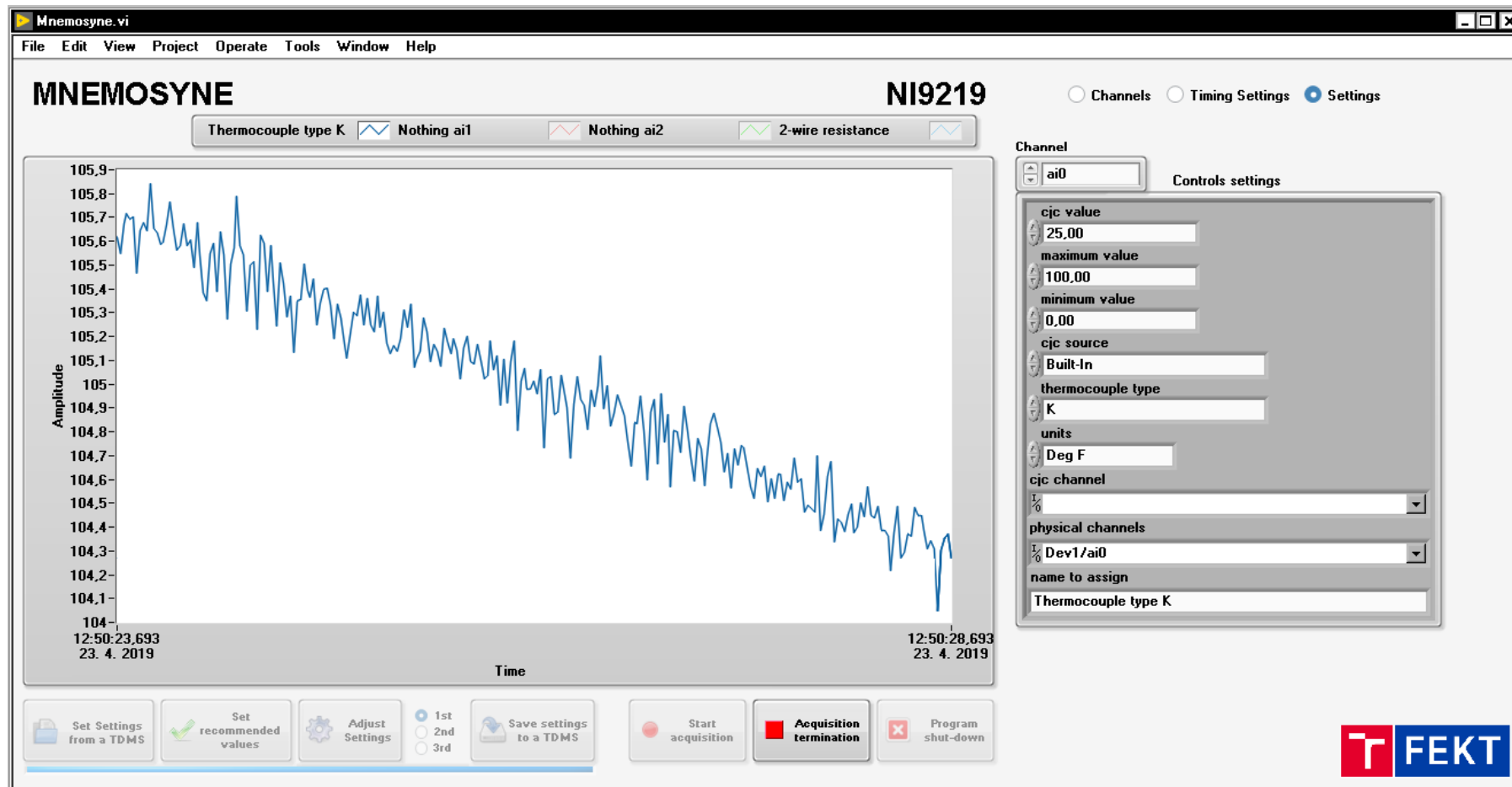




Obr. 5.3: Meranie 2-vodičového odporu (2-wire resistance) pri 10°C až 60°C (Mne-masyn)



Obr. 5.4: Meranie pomocou termočlánku (Thermocouple); (ref. 10°C až 60°C) - Mne-masyn



Obr. 5.5: Front panel počas merania termočlánkom (Thermocouple) (Mnemosyne)

## 5.3 Výstup z merania

Vzhľad čelného panelu, respektíve nastavovacej časti aplikácie môžeme vidieť na Obr.č.5.2 pre aplikáciu Mnemasyn a na Obr.č.5.5 pre Mnemosyne. Meranie dvojvodičového odporu dopadlo v oboch aplikáciách podľa očakávania rovnako. Zaznamenali sme nárast odporu v závislosti na teplote a tým sme overili, že aplikácia komunikuje s modulom NI9219 napojeným na NI USB-9162 a v oboch prípadoch namerala vizuálne rovnaký priebeh o čom svedčia vykreslené grafy TDMS súborov na Obr.č.5.3 pre Mnemasyn a na Obr.č.5.6 pre Mnemosyne.

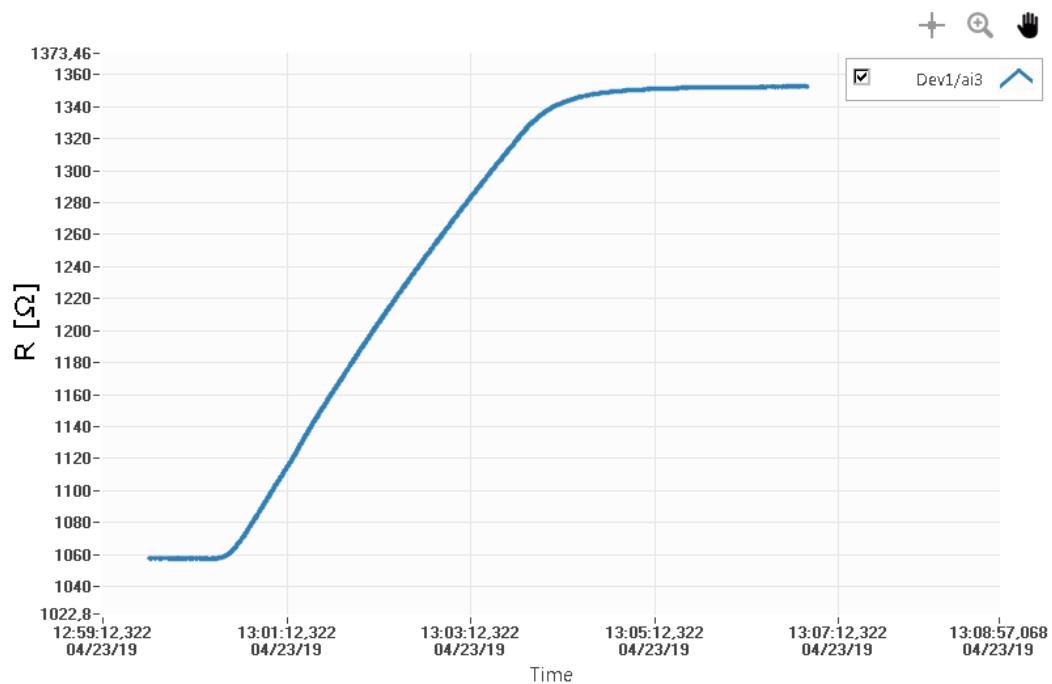
V prípade termočlánku je z Obr.č.5.4 viditeľné, že termočlánok meral na základe predpokladov. Zmenami v type termočlánkov (zmena parametru) počas merania som overil, že aplikácia reaguje na zadaný typ termočlánku. Tak isto som v priebehu merania pracoval aj s konštantnou hodnotou pre CJC<sup>1</sup> a dosahoval som rôzne výsledky - čo bolo zámerom - overenie zásahu parametru v aplikácii na výsledok. Počas merania som využil built-in zdroj pre CJC, ktorý poskytol presné výsledky a nebol dôvod pre iné riešenie čo sa týka našich potrieb vrámci tohto merania.

Na meraní z Obr.č.5.7 som použili jednotku Deg F<sup>2</sup>, takže som na grafe očakával hodnoty odpovedajúce 50°F až 140°F vychádzajúc z prevodu °C na °F ( $x9/5+32$ ), rovnako pre referenčné hodnoty 10°C až 60°C. Meraniami sme chceli dokázať funkčnosť zmeny parametrov aplikácie, schopnosť aplikácie komunikovať so zariadením NI9219 a schopnosť vykonávať merania rôznych druhov na viacerých kanáloch zároveň, čo sa týmto jednoduchým meraním potvrdilo.

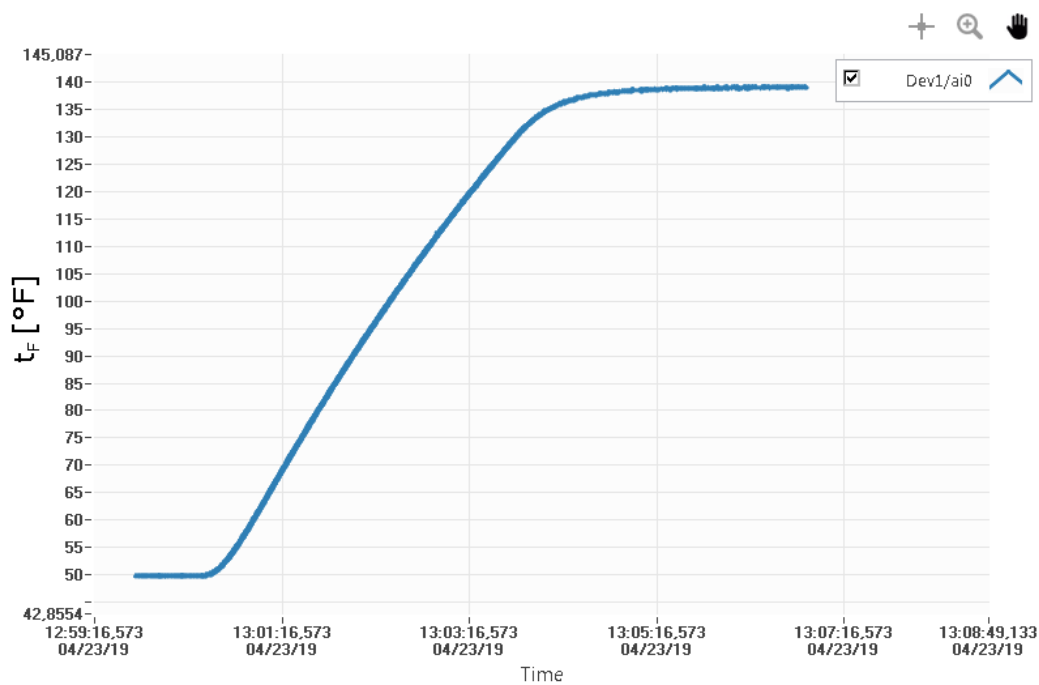
---

<sup>1</sup>Cold Junction Compensation - kompenzácia studeného konca

<sup>2</sup>Stupeň Fahrenheita; °F



Obr. 5.6: Meranie 2-vodičového odporu (2-wire resistance) pri 10°C až 60°C (Mnemosyne)



Obr. 5.7: Meranie pomocou termočlánku (Thermocouple); (ref. 10°C až 60°C) - Mnemosyne

## 6 Možné rozšírenia programu

V priebehu vypracovávania zadania mi napadlo viacero oblastí v ktorých sa aplikácia dala rozšíriť, poprípade urobiť inak. Oblasti ktoré spomeniem mňa osobne veľmi zaujímajú a v rámci pokračovania v práci by som sa im rád venoval.

### 6.1 Vzhľad

Jednoduché animačné prvky a vzhľad pripomínajúci napríklad bežnú aplikáciu Windows 10 môže byť zaujímavým extra dodatkom k takejto aplikácii, ktorý ju urobí v skutočnosti znova užívateľsky prívetivejšou a nebude vyzeráť ako tradičná LabVIEW aplikácia, ktorá by už mohla byť pre istých stálejších používateľov vizuálne neatraktívna.

### 6.2 Sprievodca nastaveniami - User Guide

Sprievodcu nastaveniami som mal pôvodne v pláne urobiť a jej implementácia je vysvetlená v sekcii 2.1.2: Sprievodca. Nakoniec na to nebol priestor z časového hľadiska, každopádne, hoci to nebolo súčasťou zadania, tak si myslím, že takáto možnosť by rozšírila túto aplikáciu správnym smerom a urobila by ju ako atraktívnejšou, tak užívateľsky prívetivejšou. Je takmer pravidlom, že špecifickejšie technické produkty sa oddiaľujú bežným užívateľom, keďže ich cieľovým publikom sú zvyčajne vzdelanejšie publikum, poprípade sú to osoby z oboru z ktorého produkt pochádza. Chyba ale takmer stále tkvie v tom, že sa nevynaložilo takmer žiadne úsilie na to, urobiť tieto produkty prístupnejšími. Z tohto dôvodu si myslím, že by bol takýto prvok veľkým obohatením výsledného produktu.

### 6.3 Zmeny pre hromadnú implementáciu

Náš cluster, poprípade pole clusterov slúžiaci viacerým kanálom sa dá rozšíriť. Aplikácia by sa dala rozšíriť na viacero typov meraní a tak isto na viacero produktov od NI. Istým druhom rozšírenia by mohla byť aplikácia, ktorá by pokrývala nielen NI9219, ale užívateľa by sa vopred opýtala s akou kartou alebo akým produktom ide pracovať a na základe toho by prispôsobila svoje možnosti. Mechanizmy s ktorými som začínal prvé verzie Mnemosyne boli veľmi jednoduché a stále som sa snažil ich mierne zovšeobecniť a verziu za verziou som si všimol, že sa to dá urobiť stále

všeobecnejším. Práve v tom vidím potenciál nájsť čo najvšeobecnejšie riešenia daných funkčných oblastí aplikácie, aby sa dali ľahko prispôbiť pre použitie s inými produktami od NI.

## 7 Záver

V prvej kapitole práce som sa venoval zorientovaniu v problematike, kapitola predstavuje teoretický úvod k danej téme. Vyjadril som sa k produktom, ktoré možno používať s kartou NI9219 a objasnil som problematiku použitia vhodného softwaru pre programovanie meracích aplikácií na zadanú meraciu kartu. V tretej sekcii som popísal vybraný software - vývojové prostredie LabVIEW, ku ktorému bolo poskytnuté odôvodnenie pre jeho výber. Štvrtá sekcia kapitoly sa venovala meracím možnostiam karty, čo v závere predstavuje šírku možností, ktoré pokrýva táto karta. Táto šírka meracieho portfólia predstavuje možnosť, ktorú sa mi podarilo sprístupniť na využitie používateľom na akomkoľvek analógovom vstupe tejto karty.

Nasledoval podrobný návrh pre zostrojenie aplikácie. Vytvoril som návrh užívateľského prostredia a návrh kódu, začínajúc s blokovými diagramami, poprípade inými možnostami pre symbolické zobrazenie žiadaných prvkov.

V tretej kapitole som sa podrobnejšie venoval dostupným kódom, konkrétne išlo o template od NI slúžiaci práve na dátovú akvizíciu a logovanie dát s názvom Continuous Measurement and Logging. Nedarilo sa mi písať kód rovno do tejto architektúry a tak som ju najprv zdokumentoval, aby som systémom v nej lepšie porozumel. Popísal som hlavne komunikačné mechanizmy ako rady a notifikátory, pričom rady hrali dôležitú rolu nositeľa dátovej komunikácie, tak aj nositeľa 3 rôznych komunikačných entít pre prenášanie riadiacich správ naprieč hlavnými slučkami programu.

V tomto úseku práce som mal maximálny teoretický základ pre tvorbu požadovanej aplikácie. Napriek tomu sa mi opakovane nedarilo implementovať svoje plány rovno do asynchrónnej architektúry (šablóna) a tak som sa rozhodol urobiť od základu vlastný kód s tým, že to bude pravdepodobne na záver ľahšie pre spomínanú implementáciu. Moja aplikácia po vypracovaní spĺňala podmienky zadania a funguje sekvenčným mechanizmom (využíva stavový automat). V skutočnosti nebol dôvod pokračovať v ďalšom vývoji, keďže by asynchrónna štruktúra nepriniesla dostatočne citelné benefity. Z dôvodu záujmu sa posunúť v téme či projekte a z dôvodu možných benefitov implementácie do známej architektúry, som vykonal aj implementáciu do šablóny. Tieto snahy skutočne viedli k dokonalejšiemu pochopeniu ako notifikátorov a rád (Queues), tak iných štruktúr LabVIEW a v závere považujem tento krok za veľmi prínosný. V tomto bode vychádzam s aplikáciou Mnemosyne ako od základu osobitým riešením zadania a aplikáciou Mnemasyn, ktorá vznikla implementáciou kódu do už existujúcej architektúry, ktorá vo všeobecnosti poskytuje väčšiu čitateľnosť a porozumiteľnosť pre iných používateľov, alebo lepšie povedané - čitateľov kódu, respektíve developerov, keďže z pohľadu užívateľa číreho výsledku (aplikácia

v podobe užívateľského prostredia) nie sú tieto dve aplikácie až tak rozdielne.

Po vytvorení aplikácií sme ich podrobili kompletnému otestovaniu v podobe simulácie. Ich funkcie ako akvizícia rôznych typov meraní, práca so súbormi alebo funkčnosť prostriedkov užívateľského prostredia boli dôkladne preverené a priestor sa venoval aj opravám chýb, na ktoré sme v tejto fázi natrafili. Po odstránení chýb sme programy otestovali na reálnom hardware - meracej karte NI9219. Týmto meraním sme chceli demonštrovať reálnu schopnosť aplikácie fungovať v praxi, pričom zámerom ale nebolo sa venovať presnosti merania, poprípade presnosti použitých meradiel či možným chybám prameniáciom z nevhodných podmienok pre meranie. Jednoduchým spôsobom sme predviedli funkcionality aplikácie a zhodnotili sme výstup z tohto merania v závere kapitoly 5.3: Výstup z merania.

Ako posledné sa dostalo priestoru možným rozšíreniam aplikácie. V kapitole sme sa venovali možnej úprave vzhľadu pre obzvláštnenie práce alebo zvýšenie užívateľskej prívetivosti. V oblasti týkajúcej sa uľahčovaniu práce užívateľom bol spomenutým prvkom „Sprievodca“ a načrtli sme aj do možného rozšírenia aplikácie na širšie portfólio produktov od spoločnosti National Instruments.



# Literatúra

- [1] National Instruments: *Datasheet NI 9219* [online]. poslední aktualizace 16. 4. 2016 [cit. 8. 11. 2018]. Dostupné z URL:  
<[http://www.ni.com/pdf/manuals/374473a\\_02.pdf](http://www.ni.com/pdf/manuals/374473a_02.pdf)>.
- [2] National Instruments: *Getting Started Guide NI 9219* [online]. poslední aktualizace 8. 12. 2017 [cit. 8. 11. 2018]. Dostupné z URL:  
<<http://www.ni.com/pdf/manuals/377223a.pdf>>.
- [3] National Instruments: *Using Traditional NI-DAQ (Legacy) and Cold Junction Compensation Sensors* [online]. poslední aktualizace 5. 1. 2018 [cit. 12. 11. 2018]. Dostupné z URL:  
<<https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000PApoSAG&l=cs-CZ>>.
- [4] National Instruments: *Measuring Strain with Strain Gages* [online]. poslední aktualizace 25. 5. 2016 [cit. 17. 11. 2018]. Dostupné z URL:  
<<http://www.ni.com/white-paper/3642/en/>>.
- [5] The Editors of Encyclopaedia Britannica: *Bridge - ELECTRICAL INSTRUMENT* [online]. poslední aktualizace 7. 2017 [cit. 17. 11. 2018]. Dostupné z URL:  
<<https://bit.ly/2Q2JJdc>>.
- [6] National Instruments: *USER GUIDE AND SPECIFICATIONS - NI USB-9219* [online]. poslední aktualizace 7. 12. 2007 [cit. 18. 11. 2018]. Dostupné z URL:  
<<http://www.ni.com/pdf/manuals/372407a>>.
- [7] National Instruments: *IEEE 1451.4 Sensor Templates Overview* [online]. poslední aktualizace 24. 8. 2016 [cit. 26. 11. 2018]. Dostupné z URL:  
<<http://www.ni.com/white-paper/3468/en/>>.
- [8] The Modal Shop - MTS Systems Corporation: *Frequently Asked Questions About TEDS* [online]. poslední aktualizace n/a ; [cit. 26. 11. 2018]. Dostupné z URL:  
<<http://www.modalshop.com/filelibrary/TEDS%20FAQ.pdf>>.
- [9] National Instruments: *Configuring a TEDS Sensor in NI MAX* [online]. poslední aktualizace 7. 12. 2018 [cit. 26. 11. 2018]. Dostupné z URL:  
<<https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019RNESA2&l=cs-CZ>>.

- [10] National Instruments: *NI 9219 Specification-eng* [online]. poslední aktualizace 2012 [cit. 29. 11. 2018]. Dostupné z URL:  
<<https://bit.ly/2Jm684C>>.
- [11] National Instruments: *C Series Module Compatibility Chart* [online]. poslední aktualizace 26. 11. 2018 [cit. 29. 11. 2018]. Dostupné z URL:  
<<http://www.ni.com/product-documentation/8136/en/>>.
- [12] National Instruments: *CompactDAQ Chassis - USB-9162* [online]. poslední aktualizace n/a ; [cit. 29. 11. 2018]. Dostupné z URL:  
<<http://www.ni.com/cs-cz/support/model.usb-9162.html>>.
- [13] National Instruments: *Frequently Asked Questions about TEDS Sensors* [online]. poslední aktualizace 6. 2. 2018 [cit. 10. 12. 2018]. Dostupné z URL:  
<<http://www.ni.com/product-documentation/8136/en/>>.
- [14] Jeff.P.Bohrer: *Re: Device vs cDAQ Module* [online]. poslední aktualizace 26. 9. 2012 [cit. 10. 12. 2018]. Dostupné z URL:  
<<https://forums.ni.com/t5/Multifunction-DAQ/Device-vs-cDAQ-Module/td-p/2168320>>.
- [15] National Instruments: *PRODUCT FLYER - CompactDAQ Chassis* [online]. poslední aktualizace 22. 9. 2017 [cit. 11. 12. 2018]. Dostupné z URL:  
<<http://www.ni.com/pdf/product-flyers/compactdaq-chassis.pdf>>.
- [16] National Instruments Corporation: *USER GUIDE AND SPECIFICATIONS - NI WLS/ENET-9163* [online]. poslední aktualizace 10. 2. 2010 [cit. 11. 12. 2018]. Dostupné z URL:  
<<http://www.ni.com/pdf/manuals/372488c.pdf>>.
- [17] National Instruments: *USER GUIDE - NI USB-9162* [online]. poslední aktualizace 13. 5. 2013 [cit. 13. 12. 2018]. Dostupné z URL:  
<<http://www.ni.com/pdf/manuals/374014a.pdf>>.
- [18] National Instruments: *C Series with Spring Terminal Front Connection: I/O Cable and Accessory Guide* [online]. poslední aktualizace 18. 4. 2018 [cit. 13. 12. 2018]. Dostupné z URL:  
<<http://www.ni.com/product-documentation/53885/en/>>.
- [19] National Instruments: *NI 9972 - Strain Relief and Operator Protection for 6-Position Modules* [online]. poslední aktualizace n/a ; [cit. 13. 12. 2018]. Dostupné z URL:  
<<http://sine.ni.com/nips/cds/view/p/lang/cs/nid/211305>>.

- [20] National Instruments: *connector type 9219* [online]. poslední aktualizace n/a ; [cit. 13. 12. 2018]. Dostupné z URL:  
<<http://sine.ni.com/nips/cds/view/p/lang/cs/nid/211278>>.
- [21] S.J. Orfanidis: *Introduction to Signal Processing* [online].  
[cit. 18. 12. 2018]. Dostupné z URL:  
<<http://eceweb1.rutgers.edu/~orfanidi/intro2sp/orfanidis-i2sp.pdf>>.
- New Jersey: Rutgers University, 2010. s. 13–20 [ Sampling and Reconstruction 1]. ISBN 0-13-209172-0.
- [22] O. Feely: *Theory of lowpass and bandpass sigma-delta modulation* [online]. [cit. 14. 12. 2018]. Dostupné z URL:  
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=494851&isnumber=10742>>. IEE Colloquium on Oversampling and Sigma-Delta Strategies for DSP, London, UK, 1995, pp. 7/1-7/8.
- [23] WEBSTER, J. G.: *The measurement, instrumentation, and sensors handbook [32.2 Resistive Thermometers; 32.4 Thermocouple Thermometers]*. Boca Raton, Fla, CRC Press published in cooperation with IEEE Press (1999). ISBN 0-8493-2145-X
- [24] ĎAĎO, Stanislav a Marcel KREIDL: *Senzory a měřicí obvody*. Praha: České vysoké učení technické, 1996. ISBN 80-01-01500-9.
- [25] National Instruments: *Cold Junction Compensation and Why It Is Needed* [online]. poslední aktualizace 3. 1. 2019 [cit. 8. 1. 2019]. Dostupné z URL:  
<<https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000P6wvSAC&l=cs-CZ>>.
- [26] Maxim Integrated: *Implementing Cold-Junction Compensation in Thermocouple Applications* [online]. poslední aktualizace 26. 4. 2007 [cit. 8. 1. 2019]. Dostupné z URL:  
<<https://www.maximintegrated.com/en/app-notes/index.mvp/id/4026>>.
- [27] Technická univerzita Ostrava: *Měření síly pomocí tenzometrů zapojených do můstku* [online]. poslední aktualizace n/a ; [cit. 12. 1. 2019]. Dostupné z URL:  
<[http://fei1.vsb.cz/kat420/vyuka/Bakalarske/STB/3\\_LU\\_tenzometry.pdf](http://fei1.vsb.cz/kat420/vyuka/Bakalarske/STB/3_LU_tenzometry.pdf)>.

- [28] UAMT FEI STU: *MERANIE MECHANICKÝCH VELIČÍN - TENZOMETRE* [online]. poslední aktualizace n/a ; [cit. 12. 1. 2019]. Dostupné z URL: <http://senzor.robotika.sk/sensorwiki/images/7/7f/Tenz.pdf>.
- [29] National Instruments: *Configuring the Digital In Threshold for the NI 9219* [online]. poslední aktualizace 6. 2010 [cit. 15. 1. 2019]. Dostupné z URL: [http://zone.ni.com/reference/en-XX/help/370984T-01/criondevicehelp/9219\\_setting\\_dithreshold/](http://zone.ni.com/reference/en-XX/help/370984T-01/criondevicehelp/9219_setting_dithreshold/).
- [30] National Instruments: *ANNUAL REPORT PURSUANT TO SECTION 13 OR 15(D) OF THE SECURITIES EXCHANGE ACT OF 1934* [online]. poslední aktualizace 31. 12. 2016 [cit. 22. 1. 2019]. Dostupné z URL: <https://www.ni.com/pdf/nati/us/10k-16.pdf>.
- [31] National Instruments: *PXI COMPONENTS* [online]. poslední aktualizace n/a [cit. 22. 1. 2019]. Dostupné z URL: <http://www.ni.com/cs-cz/shop/pxi.html>.
- [32] National Instruments: *CompactRIO COMPONENTS* [online]. poslední aktualizace n/a [cit. 22. 1. 2019]. Dostupné z URL: <http://www.ni.com/cs-cz/shop/compactrio.html>.
- [33] National Instruments: *PC-BASED MEASUREMENT AND CONTROL SYSTEM COMPONENTS* [online]. poslední aktualizace n/a [cit. 22. 1. 2019]. Dostupné z URL: <https://bit.ly/2Vtm8nc>.
- [34] National Instruments: *Programming Environments for Data Acquisition and Control* [online]. poslední aktualizace n/a [cit. 22. 1. 2019]. Dostupné z URL: <https://bit.ly/2JC0dpU>.
- [35] National Instruments: *NI LabWindows<sup>TM</sup>/CVI<sup>TM</sup> Integrated Development Environment for Test and Measurement* [online]. poslední aktualizace n/a [cit. 22. 1. 2019]. Dostupné z URL: <http://www.ni.com/lwcvl/whatis/integrated/#toc4>
- [36] National Instruments: *Measurement Studio* [online]. poslední aktualizace n/a [cit. 22. 1. 2019]. Dostupné z URL: <http://www.ni.com/mstudio/>.
- [37] National Instruments: *NI Software Portfolio* [online]. poslední aktualizace n/a ; [cit. 23. 1. 2019]. Dostupné z URL: <http://www.ni.com/cs-cz/shop/software-portfolio.html>.

- [38] EE Times, Martin Rowe: *IEEE Survey Ranks Programming Languages* [online]. poslední aktualizace 8. 9. 2018 [cit. 23. 1. 2019]. Dostupné z URL: [https://www.eetimes.com/document.asp?doc\\_id=1333572](https://www.eetimes.com/document.asp?doc_id=1333572).
- [39] National Instruments: *NI LabVIEW C Generator* [online]. poslední aktualizace n/a ; [cit. 23. 1. 2019]. Dostupné z URL: <https://sine.ni.com/nips/cds/view/p/lang/cs/nid/209015>.
- [40] Rick Bitter, Taqi Mohiuddin, Matt Nawrocki: *LabVIEW : Advanced Programming Techniques [Introduction to LabVIEW]*. CRC Press ; Taylor & Francis Group 2007 ISBN 0-8493-3325-3
- [41] Behzad Ehsani: *Data Acquisition Using LabVIEW [Introduction to LabVIEW s. 7,8]*. © 2016 Packt Publishing ISBN 978-1-78217-216-1
- [42] National Instruments, LabVIEW Help 2018: *Notifier Operations Functions, Obtain Notifier, Release Notifier, Send Notification, Want on Notification* [online]. poslední aktualizace n/a ; [cit. 16. 2. 2019]. Dostupné z URL: [https://zone.ni.com/reference/en-XX/help/371361R-01/lvhelp/labview\\_help/](https://zone.ni.com/reference/en-XX/help/371361R-01/lvhelp/labview_help/).
- [43] National Instruments, LabVIEW Help 2018: *Queue Operations Functions, Obtain Queue, Enqueue Element, Lossy Enqueue Element, Dequeue Element, Flush Queue, Release Queue* [online]. poslední aktualizace n/a ; [cit. 18. 2. 2019]. Dostupné z URL: [https://zone.ni.com/reference/en-XX/help/371361R-01/lvhelp/labview\\_help/](https://zone.ni.com/reference/en-XX/help/371361R-01/lvhelp/labview_help/).
- [44] National Instruments, LabVIEW Help 2018: *Event Structure, Create User Event, Register For Events, Generate User Event, Unregister For Events, Destroy User Event, Creating Custom Events, Create User Event - Examples* [online]. poslední aktualizace n/a ; [cit. 19. 2. 2019]. Dostupné z URL: [https://zone.ni.com/reference/en-XX/help/371361R-01/lvhelp/labview\\_help/](https://zone.ni.com/reference/en-XX/help/371361R-01/lvhelp/labview_help/).
- [45] National Instruments, LabVIEW Help 2018: *Bundle By Name, Enqueue Element At Opposite End, Merge Errors, Unbundle By Name* [online]. poslední aktualizace n/a ; [cit. 19. 2. 2019]. Dostupné z URL: [https://zone.ni.com/reference/en-XX/help/371361R-01/lvhelp/labview\\_help/](https://zone.ni.com/reference/en-XX/help/371361R-01/lvhelp/labview_help/).

- [46] National Instruments: *Top 5 LabVIEW Rookie Mistakes* [online]. poslední aktualizace n/a ; [cit. 5. 5. 2019]. Dostupné z URL:  
<<http://www.ni.com/newsletter/51735/en/>>.
- [47] National Instruments: *Continuous Measurement and Logging (NI-DAQmx) documentation* [online]. poslední aktualizace n/a ; [cit. 6. 5. 2019]. Dostupné z URL:  
<<http://www.ni.com/product-documentation/14031/en/#toc5>>.
- [48] National Instruments: *User Guide NI USB-9162* [online]. poslední aktualizace n/a ; [cit. 6. 5. 2019]. Dostupné z URL:  
<<http://www.ni.com/pdf/manuals/374014a.pdf>>.
- [49] National Instruments: *NI-9219, C Series Universal Analog Input Module* [online]. poslední aktualizace n/a ; [cit. 14. 5. 2019]. Dostupné z URL:  
<<http://www.ni.com/cs-cz/support/model.ni-9219.html>>.

# Zoznam symbolov, veličín a skratiek

<b>AI</b>	Analog Input
<b>ADC</b>	Analog-to-Digital Converter - analogovo digitálny prevodník
<b>CJC</b>	Cold Junction Compensation - kompenzácia studeného konca
<b>DAQ</b>	Data Acquisition
<b>DBL</b>	Double -> Double precision floating-point format
<b>EEPROM</b>	Electrically Erasable Programmable Read-Only Memory - elektronicky mazateľná pamäť, programovateľná na využitie iba na čítanie
<b>FPGA</b>	Field-Programmable Gate Array - Programovateľné hradlové pole
<b>GUI</b>	Graphical User Interface - grafické užívateľské rozhranie
<b>HW</b>	Hardware - fyzické vybavenie výpočetnej techniky
<b>IEC</b>	International Electrotechnical Commission
<b>IEEE</b>	Institute of Electrical and Electronics Engineers - Inštitút pre elektrotechnické a elektronické inžinierstvo
<b>LabVIEW</b>	Laboratory Virtual Instrument Engineering Workbench
<b>MCS</b>	Main Controls Settings - array clusterov s hlavnými nastaveniami aplikácie Mnemasyn
<b>NI</b>	National Instruments
<b>OS</b>	Operating System - operačný systém
<b>PCI</b>	Peripheral Component Interconnect
<b>PXISA</b>	PXI Systems Alliance
<b>RTD</b>	Resistance Temperature Detector - Odporový teplomer/Odporový teplotný detektor
<b>TDMS</b>	Technical Data Management System - Systém uchovávaní a radenia informácií v dokumente
<b>TEDS</b>	Transducer Electronic Data Sheet - štandardizovaná metóda uchovávaní informácií o senzore
<b>UI</b>	User Interface - užívateľské prostredie
<b>USB</b>	Universal Serial Bus - Univerzálna sériová zbernica
<b>VI</b>	Virtual Instrument

## Obsah priloženého CD

V priloženom CD sa nachádzajú kompletne zdrojové kódy pre aplikáciu Mnemosyne a Mnemasyn. Ako pri vývoji, tak pri testovaní a simulácii bol použitý program LabVIEW 2018, verzia 18.0f (32-bit).

Na CD nosiči sa ďalej nachádza „MnemasynDemo.wmv“ (video) ilustrujúce použitie aplikácie Mnemasyn z užívateľského hľadiska. „MnemasynDemoEng.wmv“ je jeho kópiou v anglickom jazyku. Súbor „Pizur\_BP\_2019.pdf“ predstavuje hlavný dokument písomnej časti bakalárskej práce. V následovnej hierarchickej štruktúre sú spomenuté kľúčové súbory, zložky a cesta k nim. Obsah priloženého CD obsahuje súborov viac, táto štruktúra posluží na zorientovanie sa v súboroch pre nás podstatných.

```
/.....cesta k zložke priloženého CD
├── Mnemasyn.....aplikácia Mnemasyn so všetkými náležitostami
│   ├── Mnemasyn.vi.....VI aplikácie Mnemasyn
│   ├── Mnemasyn.lvproj.....projekt aplikácie
│   └── Builds.....skompilované inštancie programu a inštalačné súbory
│       └── Build4.....posledná verzia
│           ├── Mnemasyn.exe.....samostatná spustiteľná aplikácia Mnemasyn
│           └── Installer.....priestor pre „Build“ inštalačného súboru
├── Mnemosyne.....aplikácia Mnemosyne so všetkými náležitostami
│   ├── Mnemosyne.vi.....VI aplikácie Mnemosyne
│   ├── Mnemosyne.lvproj.....projekt aplikácie
│   └── Builds.....skompilované inštancie programu a inštalačné súbory
│       └── Build3.....posledná verzia
│           ├── Mnemosyne.exe.....samostatná spustiteľná aplikácia Mnemosyne
│           └── Installer.....priestor pre „Build“ inštalačného súboru
├── MnemasynDemo.wmv.....demo Mnemasyn Sk
├── MnemasynDemoEng.wmv.....demo Mnemasyn Eng
└── Pizur_BP_2019.pdf.....písomná časť bakalárskej práce
```